AÑO III - NUM. 63

MICROHOBBY

SEMANAL

REVISTA INDEPENDIENTE PARA USUARIOS DE ORDENADORES SINCLAIR

AÑO III - N.° 63

135 PTS.
Incluido IVA



PROFESOR PARTICULAR

REPRESENTACIÓN GRÁFICA DE FUNCIONES

INICIACION

MAS POSIBILIDADES SONORAS PARA EL 128 K

NUEVO

¡OLÉ, TORO!

VIVE LA EMOCIÓN DE UNA AUTÉNTICA CORRIDA

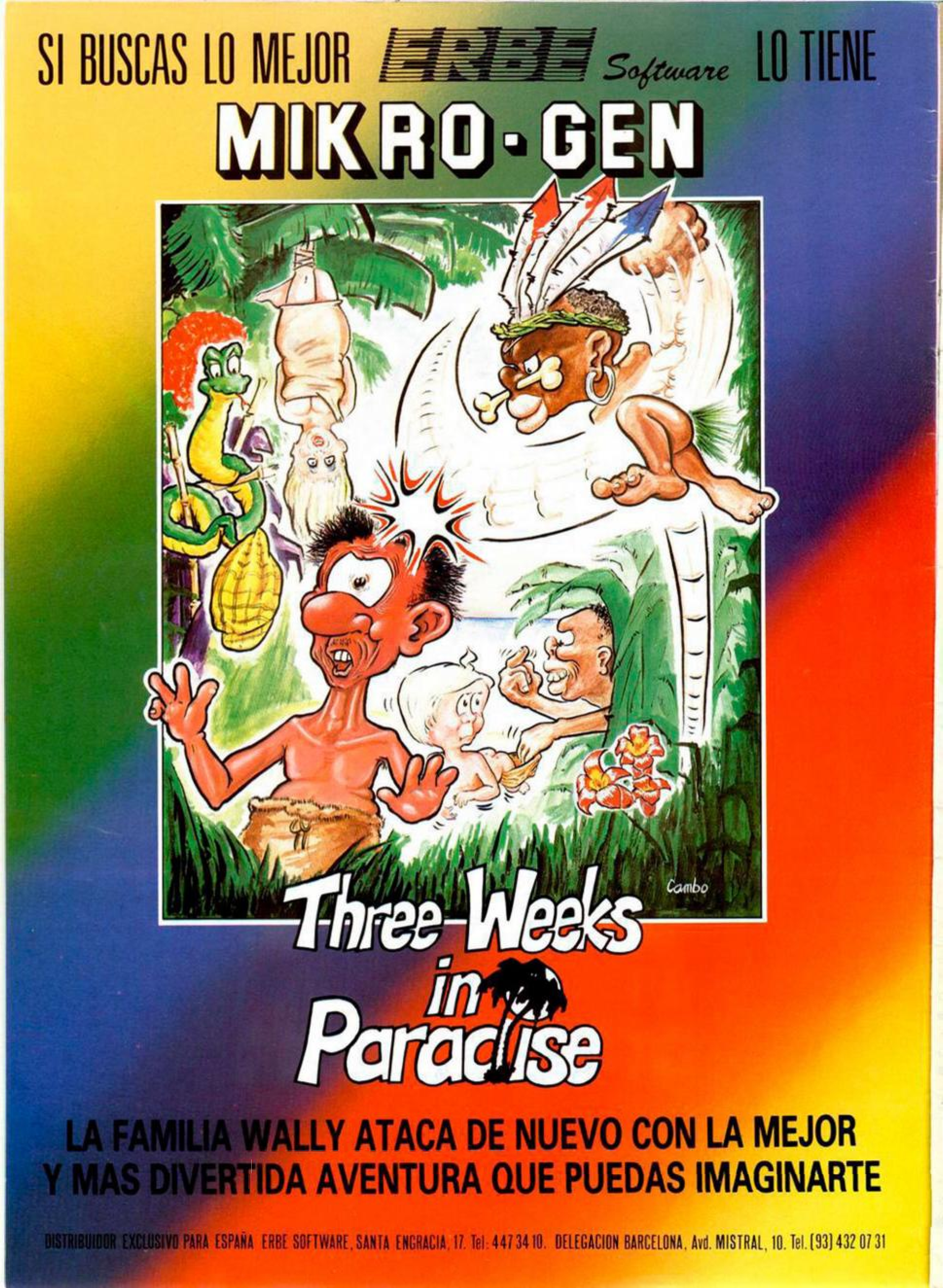
HARDWARE

COMO USAR LAS INTERRUPCIONES EN EL Z 80

TRUCOS

GRÁFICOS PARA TODOS LOS GUSTOS

HOBBY PRESS, S.A.


SI BUSCAS LO MEJOR
ERBE Software
LO TIENE
MIKRO-GEN
Cambo
Three Weeks in Paradise
LA FAMILIA WALLY ATACA DE NUEVO CON LA MEJOR
Y MAS DIVERTIDA AVENTURA QUE PUEDAS IMAGINARTE
DISTRIBUIDOR EXCLUSIVO PARA ESPAÑA ERBE SOFTWARE, SANTA ENGRACIA, 17. Tel: 447 34 10. DELEGACION BARCELONA, Avd. MISTRAL, 10. Tel. (93) 432 07 31

Director Editorial
José I. Gómez-Centurión
Director Ejecutivo
Domingo Gómez
Asesor Editorial
Gabriel Nieto
Redactor Jefe
Africa Pérez Tolosa
Diseño
Rosa María Capitel
Redacción
Amalio Gómez, Pedro Pérez, Jesús Alonso
Secretaria Redacción
Carmen Santamaría
Colaboradores
Primitivo de Francisco, Rafael Prades, Miguel Sepúlveda Sergio Martínez y J. M. Lazo
Corresponsal en Londres
Alan Heap
Fotografía
Javier Martínez, Carlos Candel
Portada
José María Ponce
Dibujos
J. R. Ballesteros, A. Perera, F. L. Frontán, Pejo, J. M. López Moreno, J. Igual, J. A. Calvo, Lóriga, J. Olivares

Edita
HOBBY PRESS, S. A.
Presidente
María Andrino
Consejero Delegado
José I. Gómez-Centurión
Jefe de Publicidad
Marisa Esteban
Publicidad Barcelona
José Galán Cortés
Tels.: 303 10 22 - 313 71 76
Secretaria de Dirección
Marisa Cogorro
Suscripciones
M.ª Rosa González
M.ª del Mar Calzada
Redacción, Administración y Publicidad
La Granja, 39
Polígono Industrial de Alcobendas
Tel.: 654 32 11
Telex: 49480 HOPR
Dto. Circulación
Carlos Peropadre
Distribución
Coedis, S. A. Valencia, 245
Barcelona
Imprime
Rotedic, S. A. Ctra. de Irún, km. 12,450 (MADRID)
Fotocomposición
Espacio y Punto, S. A.
Paseo de la Castellana, 268
Fotomecánica
Grof
Ezequiel Solana, 16
Depósito Legal
M-36.598-1984

Representante para Argentina, Chile, Uruguay y Paraguay, Cía. Americana de Ediciones, S.R.L. Sud América 1.532. Tel.: 21 24 64. 1209 BUENOS AIRES (Argentina).

Solicitado control
OJD

**Precio para Canarias, Ceuta y Melilla: 130 ptas.
Sobretasa aérea: 10 ptas.**

MICROHOBBY ESTA SEMANA

**AÑO III. N.º 63. 28 de enero al 3 de febrero de 1986
135 ptas.**





«Ole, toro», la fiesta nacional vista por los muchachos de Dinamic. Pag. 12.

PREMIADOS HOBBY-SUERTE

JUAN TAMARGO GUTIERREZ. Sta. Laura, 1, 7.º D. Móstoles (MADRID).
Cinta de programas (5.ª Cat.)

JORGE VENERO ALONSO. Doce de Octubre, 6 (MADRID).
Suscripción a Microhobby Semanal por un año (4.ª Cat.)

MIGUEL CABALLERO GARCIA. Maestro Nicolau, 14. BARCELONA.
Cinta de programas (5.ª Cat.)

JOAQUIN LA CRUZ. Avda. Roma, 127, 1.º Rexonch-Lisboa (PORTUGAL).
Impresora Seikosha (2.ª Cat.)

JOSE M.ª CORONADO TORDESILLAS. Juan II, 4, 3.º B (CIUDAD REAL).
Cinta de programas (5.ª Cat.)

ANA BELEN FELTRAN ESPEJO. Florida Blanca, 12, 3.º D. Dolores (CARTAGENA)
Impresora Seikosha (2.ª Cat.)

VICTOR LEON LAGOA. Pza. de España, 18, 2.º D. Zafra (BADAJOZ).
Cinta de programas (5.ª Cat.)

MANUEL ADRADOS ANTORALZ. Castrillo de Aza, 13 (MADRID).
Cinta de programas (5.ª Cat.)

ANTONIO NAVARRO DEL PINO. Avda. del Mediterráneo, 21, 5.º A (ALMERIA).
Cinta de programas (5.ª Cat.)

DANIEL GARCIA DURAN. Avda. Ntra. Sra. Valvanera, 104. (MADRID).

ALBERTO ACUÑA AYOZA. Archiduque Luis Salvador, 90, 2.º, 5.º (PALMA DE MALLORCA).
Suscripción a Microhobby Semanal por un año (4.ª Cat.)

JOSE ANGEL AGUILAR PEREZ. Fray Antonio de Marchena, 38. Linares (JAEN).
Cinta de programas (5.ª Cat.)

MARIANO GARICA-SALAMANCA PALAU. Marroquina, 24, 6.º D. (MADRID).
Cinta de programas (5.ª Cat.)

ELOY SERRANO BENITRAGO. Cuesta de Hista/Nuevo Alamín, 7, 5.º A (GUADALAJARA).
Suscripción a Microhobby Semanal por un año (4.ª Cat.)

MIGUEL ANGEL ALONSO DEL PINO. Fernando Caleadillo, 19 (BADAJOZ).
Cinta de programas (5.ª Cat.)

CONCEPCION SANCHEZ TORRES. Amargura, s/n. Yebra (GUADALAJARA).
Un Spectrum 48 K (1.ª Cat.)

MICROPANORAMA

EL FRACASO DEL MIKRO-PLUS

El director de Mikro-Gen, una de las compañías de software con más repercusión en el mercado, ha anunciado que no volverá a realizar ningún otro juego utilizando el Mikro-Plus, invento éste que fue usado por primera vez en el juego Shadow of the Unicorn, y que como sabréis, consiste en una ampliación de memoria de 16K.

Esta decisión ha sido tomada tras comprobar el bajo nivel de ventas alcanzado por la Sombra del Unicornio, que según el propio Mike Meek, ha sido motivada principalmente por el elevado precio con que éste salió al mercado y que lo hacía casi prohibitivo para la gran mayoría de los usuarios.

Según los cálculos de esta compañía, para que uno de sus programas se considere de éxito debe alcanzar la cifra de 20.000 libras, y por tanto este Shadow of the Unicorn debería haber llegado hasta las 40.000 libras, pero tan sólo ha conseguido vender 11.000.



Todas estas circunstancias han motivado una gran crisis en Mikro-Gen, ya que Andrew Laurie, el principal programador de la compañía y uno de sus primeros fundadores, ha sido despedido de la misma, así como el director de marketing Paul Denial.

El Mikro-plus, que prometía ser el gran éxito del año, ha acabado siendo un rotundo desastre.

IMPRESORA OFICIAL PARA EL QL

Tras un largo período de decisiones y estudio, Sinclair Research ha decidido el modelo de impresora que configurará el paquete junto con el QL.

Se trata de una máquina de impresión matricial por puntos de la marca Seikosha y que podrá ser conectada directamente al QL sin necesidad de interface. Su precio en Gran Bretaña será de unas 250 libras (cerca de 50.000 ptas.).

Alison Maguire, director de marqueting de Sinclair Research ha manifestado que han elegido esta impresora por creer que es la que posee una mejor combinación de cualidades. Estas son: velocidades de 100 caracteres por segundo, o de 25 cps para precisión de carta y opciones de alimentación de papel suelto o papel continuo.

Sin embargo, el precio que alcanzará este paquete sigue siendo superior al de uno de sus principales rivales, el procesador de textos de Amstrad, pero en todo caso la inclusión de esta nueva impresora puede darle un impulso considerable a las ventas del QL.

AQUI LONDRES

Commodore está a punto de cerrar su fábrica en Corby con la consiguiente pérdida de 250 puesto de trabajo. Esto significa que la compañía deja de tener facilidades de fabricación en el Reino Unido. El cierre deja el futuro de los ordenadores C-16 y el Plus-4 muy oscuro ya que ambos son manufacturados exclusivamente en esta ciudad. El C-64 y el 128 que se ensamblan en Corby, probablemente se fabricarán en su totalidad en la fábrica alemana de Commodore.

El cierre no afectará la producción de CPC10 y PC20 ni de Amiga ya que no se intentaba ensamblar estos en Corby.

Siguiendo el éxito de la cinta de compilación de Softaid, que alcanzó más de 350.000 £ para las víctimas del hambre en Africa, G.O.S.H va a lanzar una nueva cinta de compilación que se llamará Off The Hook. Los beneficios de esta nueva cinta serán destinados a la lucha contra el abuso de droga.

Activision ha anunciado un nuevo conjunto de software para el C-64, lo que os habilita a crear vuestros propios juegos de arcade. «Game Maker» os facilita un juego de herramientas, incluyendo diseñadores de sprites y pantallas, generador de sonido y editor de música.

El programa podrá adquirirse en marzo y costará 10 $, en cassette o 15 en disquette.

«Screen Micro» ha lanzado una ampliación de memoria para el Amstrad 464 lo que facilita a sus usuarios el uso del software para el 6128 e incluso software CPM. Por 100 £ puedes conseguir este sistema de expansión de 64K RAM que actúa como el 6128. Incluye también software que te posibilita para usar cualquier conjunto CPM 2,2 en tu ordenador.

La feria de Amstrad Computer tuvo lugar los días

Contra la piratería

DEMASIADA PROTECCION

La lente diseñada por ASAP Developments para la protección de programas, la cual supone un serio problema para los piratas del software, parece que está causando las críticas del público de Gran Bretaña.

Estos inconvenientes no vienen motivados por la lente en sí, sino por los manuales explicativos que las casas de software están dando a sus usuarios, pues resultan, según las numerosas quejas recibidas, absolutamente ambiguos y de difícil comprensión.

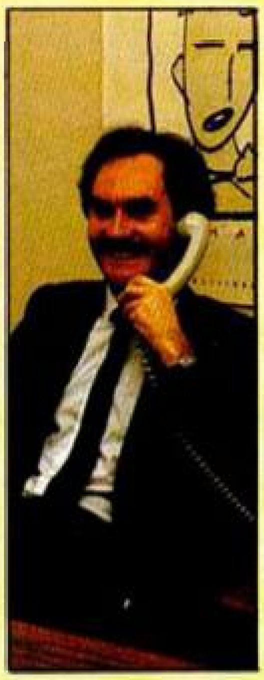
Firebird fue la primera en llevar a la práctica este revolucionario método de protección de programas con su juego Elite, pero parece que otras compañías se están animando a llevarlo a cabo con sus productos, como Digital Integration que lo aplicará en Tomahawk y posteriormente en todos sus programas.

ASAP por su parte adquirirá las instrucciones de Digital como standart, pues considera que son las que hacen más asequible al usuario su comprensión.

Parece que con esta Lenslok, las copias piratas de software tienen sus días contados.

11 y 12 de enero en el Novotel de Londres. Más de 70 compañías se exhibieron en dicha feria, muchas de las cuales presentaron sus nuevos productos tanto de software como de hardware. 1985 ha sido un año de éxito para Amstrad: empezó con el lanzamiento del CPC464 y 664 seguido por el CPC6128 y PCW8256, lo cual colocó a Amstrad como uno de los mejores productores de microordenadores en el U.K. y Europa.

Con un número creciente de sus micros, periféricos y software en el mercado, este año también promete éxitos.



De nuestro corresponsal en Londres ALAN HEAP

LIBROS



MICROINFORMATICA DE GESTION

Alternativas y Utilización

G. Guerin/Editorial Gustavo Gili/214 págs.

Este libro está especialmente destinado a aquellas personas que deben enfrentarse con la elección y uso de microordenadores en el ámbito de la gestión, para uso propio o empresarial.

La obra se divide básicamente en cuatro capitulos. El primero es un exhaustivo análisis que nos permite enfrentarnos con los problemas que acarrea todo tipo de información.

Descripción de los periféricos más usuales. Análisis de por qué es necesario un ordenador y la búsqueda de una productividad real del mismo.

El siguiente capitulo trata de los programas de aplicación y los criterios ergonómicos, técnicos y económicos que deben seguirse para su elección.

A continuación aborda los programas de «utilidad» para personas que desconocen la informática, tales como tratamientos de textos, gestión de fichas, manipulación de tablas y cálculos. Finaliza este apartado comentando los bancos de datos y su consulta.

Otro de los temas tratados es cómo realizar una acertada elección de materiales: pantalla, unidad central, impresora y equipo lógico (software) básico.

Análisis de los dos sitemas operativos por excelencia: UNIX y CP/M, ventajas e inconvenientes de cada uno de ellos.

Resumen de los lenguajes de programación más empleados, con una breve descripción de los mismos (aunque con más detenimiento en el apartado de BASIC).

Por último trata de los Programas de Ensayos comparativos de microordenadores (los llamados MIX), incluyendo una tabla de los resultados obtenidos con diversos equipos.

AHORA SÍ

puedes aprender a programar en basic de una vez por todas

¡Solicítalo antes de que se agote!
Hay un número limitado de ejemplares

DEJATE de complicados e incomprensibles sistemas de aprendizaje. Conoce de una vez por todas lo que es el Basic. Es más sencillo de lo que crees, porque ahora tienes algo que estabas esperando hace mucho tiempo: MICROBASIC, una edición corregida y revisada del famoso curso publicado por MICROHOBBY SEMANAL.

MICROBASIC es el libro que te enseñará a ser un experto en programación. Aunque hasta ahora sólo hayas utilizado tu Spectrum para jugar.

MICROBASIC te introducirá, paso a paso, en el Basic. Con ejemplos claros, sencillos y prácticos que irán adquiriendo complejidad según vayas aumentando tu nivel. Hasta llegar a dominarlo por completo.

Aprovecha esta oportunidad, porque ahora si puedes llegar a conocer a fondo tu Spectrum. Ahora, por fin, a tu alcance el método más claro y completo de programación en Basic publicado hasta el momento.



Recorta o copia este cupón y envíalo a HOBBY PRESS, S. A. Apartado de Correos 232. Alcobendas (Madrid)

Nombre __________
Apellidos __________
Dirección __________
Localidad ______ Provincia ______
Código Postal _____ Edad _____ Teléfono _____

Deseo recibir en mi domicilio el libro MICROBASIC, al precio de 1.750 ptas. (IVA incluido). El importe lo pagaré:

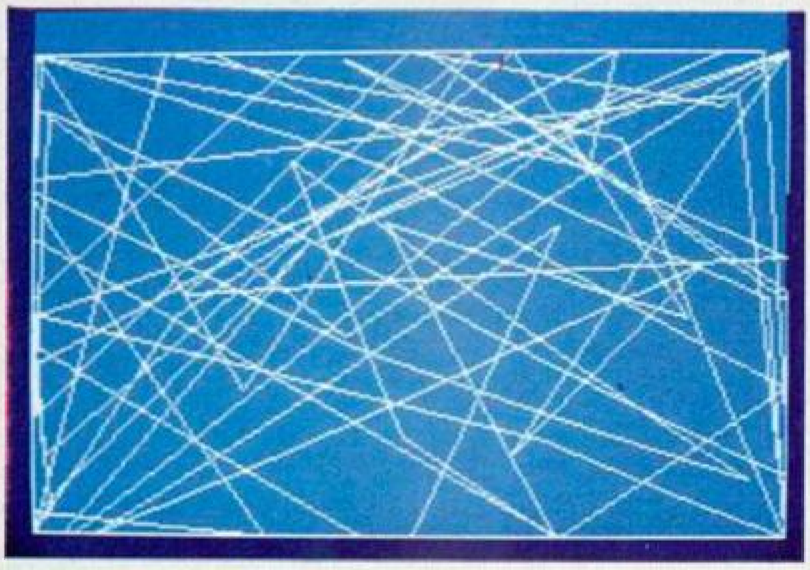
- ☐ Mediante talón bancario adjunto a nombre de HOBBY PRESS, S. A.
- ☐ Mediante tarjeta de crédito ______
  Número de la tarjeta ______
  Fecha de caducidad de la tarjeta ______
- ☐ Mediante giro postal n.° ______
- ☐ Contra reembolso (supone 75 ptas. de gastos de envío)

Fecha y firma

TRUCOS

RAYAS

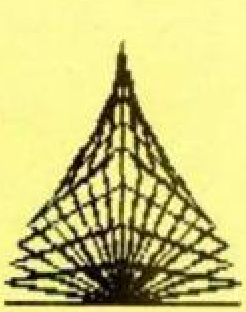
Jesús Ezquerra nos envía este truco con el que, como podéis apreciar por la foto que os adjuntamos, se pueden hacer todo tipo de combinaciones «rayadas».



```
 10 LET x=255-INT (RND*511)
 20 LET y=175-INT (RND*351)
 30 LET vx=PEEK 23677
 40 LET vy=PEEK 23678
 50 IF x+vx>255 THEN LET x=255-
vx
 60 IF x+vx<0 THEN LET x=-vx
 70 IF y+vy>175 THEN LET y=175-
vy
 80 IF y+vy<0 THEN LET y=-vy
 85 IF INKEY$<>"" THEN RUN
 90 DRAW x,y
100 GO TO 10
```

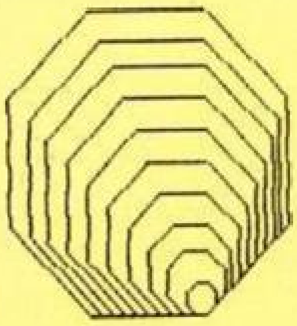
FIGURAS

Y siguiendo con las posibilidades artísticas del Spectrum, aquí os mostramos unas cuantas que pueden serviros para enriquecer vuestros programas.



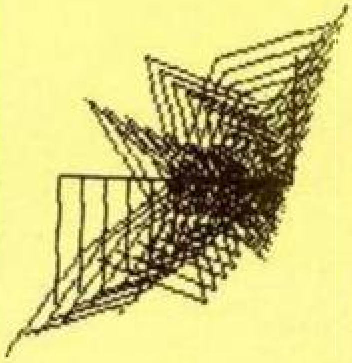
```
10 INPUT " lados ";n
20 LET n=n/2
30 FOR j=4 TO 40 STEP 4
40 FOR i=0 TO 2*PI STEP PI/n
50 PLOT 100-40*COS i,40*SIN i
60 DRAW j*COS i,j*SIN i
70 NEXT i
80 NEXT j
```

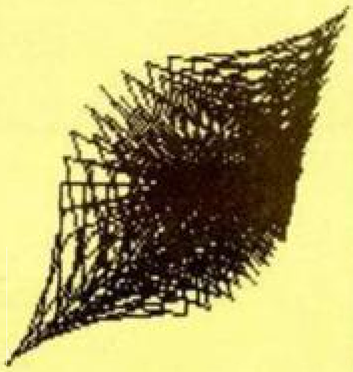
Con tan sólo añadir la línea 35 al listado, conseguiremos esto:

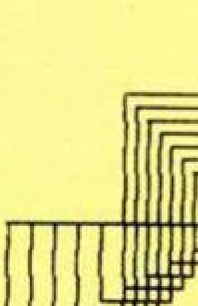


```
35 PLOT 100-j,40
```

Si queremos más complejidad sólo tendréis que sustituir las líneas 50 y 60 por las que os damos:

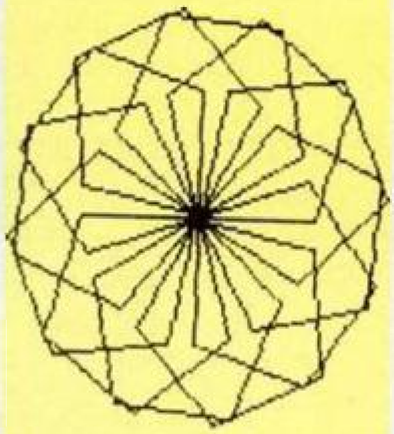


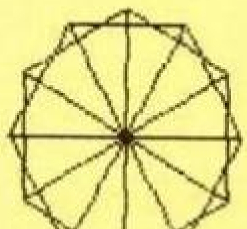




```
50 PLOT 100-j,80
65 DRAW j*SIN i
,j*COS i
```

Y, finalmente, conseguiréis estos otros con un nuevo listado.





```
10 INPUT n
20 INPUT m
30 LET m=m/2: LET n=n/2
40 FOR j=0 TO 2*PI STEP PI/m
50 PLOT 100,80
60 FOR i=0 TO 2*PI STEP PI/n
70 DRAW 40*COS (i*j),40*SIN (i
*j)
80 NEXT i
90 NEXT j
```

En este espacio también tienen cabida los trucos que nuestros lectores quieran proponer.
Para ello, no tienen más que enviarlos por correo a MICROHOBBY, C/ La Granja, 8. Polígono Industrial de Alcobendas (Madrid).

IGLOO

M. A. GOMEZ FLORES

TODAS LAS LETRAS MAYUSCULAS SUBRAYADAS DEBERAN TECLEARSE EN MODO GRAFICO.

Spectrum 48 K

Después de un frío invierno nuestro amigo, el esquimal Lemmy, sale de su igloo a la pesca de algún que otro pez para reponer sus fuerzas.

En este gélido marco, hemos de ayudarle a realizar su pesca y transportarla a su igloo teniendo en cuenta que sólo podemos coger un pez cada vez, esquivando los agujeros del hielo y a los pinguinos, unos porque nos mandarán al inicio del juego y, otros, porque nos pueden robar la pesca conseguida.

Las teclas de movimiento son:

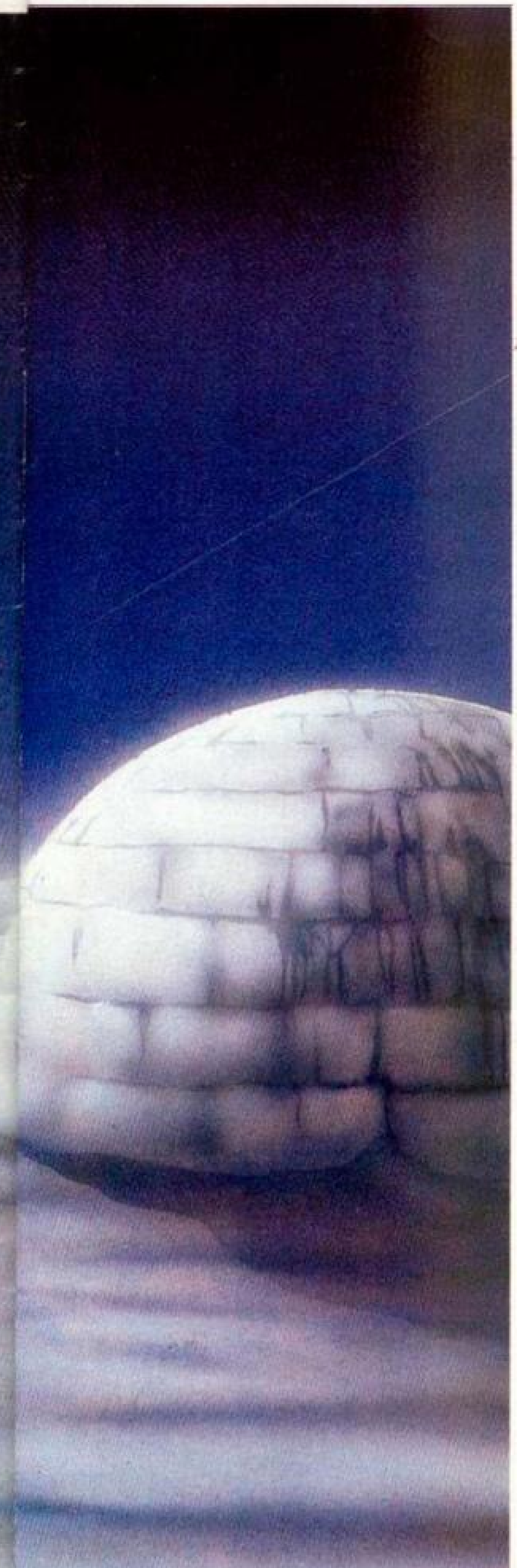
P, arriba
O, abajo
Q, derecha
A, izquierda




```
   1 BORDER 7: PAPER 7: INK 0: C
LS : PRINT FLASH 1;AT 10,10;" PA
RE LA CINTA ": FOR I=1 TO 64: BE
EP .02,I: NEXT I: CLS
   2 REM [MIGUEL ANGEL GOMEZ]
   3 REM
  10 BRIGHT 0: BORDER 3: PAPER 5
: INK 2: CLS
  15 LET re=0: POKE 23658,0: POK
E 23609,10
  20 GO SUB 8500
  30 GO SUB 5000
  40 GO TO 180
  79 REM  Le PiLLA eL pINGUINO
  80 BEEP .001,20: BEEP .01,-30:
 LET pe=0: IF pu>50 THEN LET pu=
pu-50
  90 RETURN
  99 REM  Se AhOgA eN eL hIeLO
 100 FOR n=-20 TO -10: BEEP .01,
n: NEXT n
 110 FOR n=0 TO 5: PRINT INK n;A
T yy,xx;a$;AT yy+1,xx;b$;AT a,b;
 INK n;c$: PAUSE 10: NEXT n: FOR
 n=50 TO 10 STEP -.3: BEEP .001,
n: NEXT n: GO SUB 9500: GO TO 18
0
 119 REM  CoGe Un PeZ
 120 BEEP .001,30: BEEP .01,50:
LET pe=0: PRINT AT 20,30-LEN p$;
 BRIGHT 1; PAPER 4; INK 2;p$: LE
T p$=p$+"D": LET pu=pu+100: IF L
EN p$=11 THEN GO SUB 4500: GO TO
 180
 130 RETURN
 179 REM  PoS. iNiCiAL
 180 BEEP .01,-2: PRINT INK 0;AT
 aa,bb;c$
 190 PAUSE 10: BEEP .001,30: BEE
P .002,25: PRINT INK 2;AT yy,xx;
a$;AT yy+1,xx;b$
 195 PRINT AT 18,30-LEN STR$ re;
 BRIGHT 1; PAPER 4; INK 7;re;AT
20,9;ni
 199 LET in=0
 200 REM  MoViMiENTO
 210 LET a=PEEK 23560: IF NOT a
THEN GO TO 320
 215 PRINT AT 18,15-LEN STR$ pu;
 PAPER 4; BRIGHT 1; INK 7;pu: LE
T pu=pu+1
 220 IF a=112 AND ATTR (y,x+1)<>
40 AND ATTR (y+1,x+1)<>40 THEN L
ET x=x+(x<29): LET a$="L": LET b
$="N"
 230 IF a=111 AND ATTR (y,x-1)<>
40 AND ATTR (y+1,x-1)<>40 THEN L
ET x=x-(x>2): LET a$="L": LET b$
="M"
 240 IF a=113 AND ATTR (y-1,x)<>
40 THEN LET y=y-(y>2): LET a$="Q
": LET b$="P"
 250 IF a=97 AND ATTR (y+2,x)<>4
0 THEN LET y=y+(y<15): LET a$="L
": LET b$="O"
 260 IF ATTR (y,x)=41 OR ATTR (y
+1,x)=41 THEN GO TO 100
 270 IF ATTR (yy,xx)=40 OR ATTR
```



JAVIER IGUAL

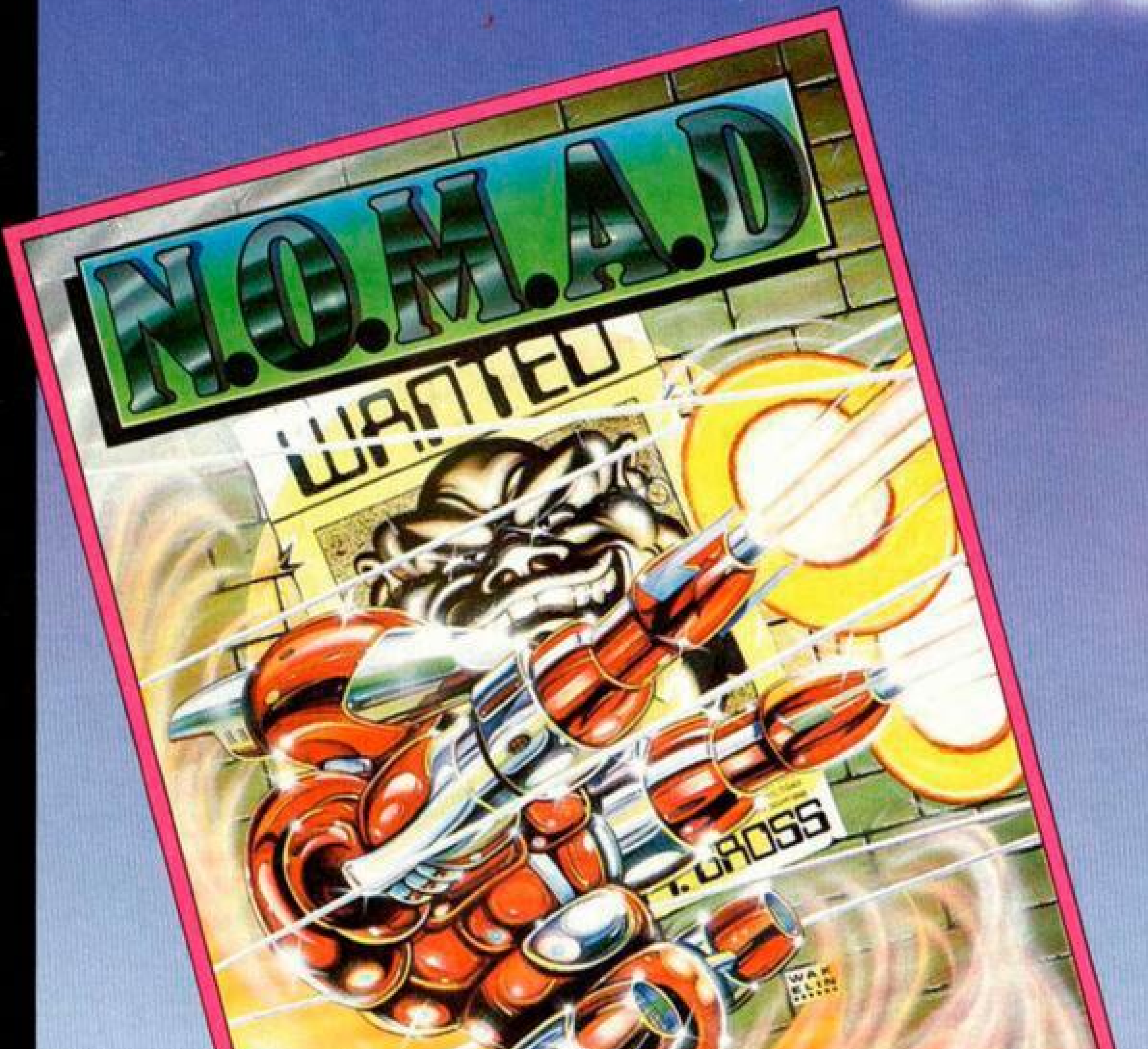


```
(yy+1,xx)=40 THEN GO SUB 80
 280 IF ATTR (y,x)=43 OR ATTR (y
+1,x)=43 THEN LET pe=1: BEEP .01
,60: BEEP .001,20: LET pu=pu+10
 290 IF y=4 AND x=9 AND pe THEN
GO SUB 120
 300 PRINT AT yy,xx;" ";AT yy+1,
xx;" "
 310 PRINT INK 2;AT y,x;a$;AT y+
1,x;b$
 320 IF a<y+2 AND ATTR (a+1,b)=4
2 THEN LET a=a+.7: LET c$="C"
 330 IF a>y AND ATTR (a-1,b)=42
THEN LET a=a-.7: LET c$="C"
 340 IF b<x AND ATTR (a,b+1)=42
THEN LET b=b+.7: LET c$="B"
 350 IF b>x AND ATTR (a,b-1)=42
THEN LET b=b-.7: LET c$="A"
 370 PRINT AT aa,bb;" ": PRINT I
NK 0;AT a,b;c$
 375 IF m=32 THEN GO SUB 9500
 380 LET yy=y: LET xx=x: LET aa=
a: LET bb=b
 400 GO TO 210
 999 REM  NiVeL 1
1000 PRINT INK 1;AT 5,14;d$;AT 7
,22;d$;AT 10,6;d$;AT 12,24;d$;AT
 13,14;d$
1090 RETURN
1099 REM  NiVeL 2
1100 PRINT INK 1;AT 3,22;d$;AT 7
,10;d$;AT 7,24;d$;AT 9,16;d$;AT
10,4;d$;AT 13,10;d$;AT 13,22;d$
1190 RETURN
1199 REM  NiVeL 3
1200 PRINT INK 1;AT 5,16;d$;AT 6
,10;d$;AT 7,4;d$;AT 7,25;d$;AT 8
,22;d$;AT 10,10;d$;AT 12,12;d$;A
T 12,18;d$;AT 13,25;d$;AT 14,3;d
$
1290 RETURN
3999 REM  MuSiCa
4000 FOR n=20 TO 22: FOR m=25 TO
 0 STEP -2: BEEP .05,n: BEEP .03
,m: BEEP .02,n+m: NEXT m: NEXT n
: FOR n=1 TO 2: BEEP .1,15: BEEP
 .2,23: NEXT n
4110 CLS : RETURN
4499 REM  MuSiCa 2
4500 RESTORE 4510: FOR m=1 TO 9:
 READ a,b: BEEP a,b: NEXT m: LET
 ni=ni+1: GO SUB 9300: IF ni=4 T
HEN GO TO 9400
4510 DATA .083,4,.084,3,.083,2,.
25,1,.25,9,.375,12,.125,14,.1,11
,.30,7
4520 CLS
4999 REM  PaNTaLLa
5000 FOR n=0 TO 16: PRINT INK 3;
 BRIGHT 0;AT n,0;"█";AT n,31;"█"
: NEXT n
5010 PLOT 0,0: DRAW 255,0: DRAW
0,39: DRAW -8,0: DRAW 0,136: DRA
W -239,0: DRAW 0,-136: DRAW -8,0
: DRAW 0,-39
5020 PLOT 7,7: DRAW PAPER 2;241,
0: DRAW PAPER 2;0,9: DRAW PAPER
2;-241,0: DRAW PAPER 2;0,-9
5030 PLOT 7,23: DRAW PAPER 2;241
,0: DRAW PAPER 2;0,9: DRAW PAPER
 2;-241,0: DRAW PAPER 2;0,-9
5040 PLOT 15,39: DRAW 225,0: DRA
W PAPER 2;0,129: DRAW PAPER 2;-2
25,0: DRAW PAPER 2;0,-129
5050 PRINT AT 1,2; PAPER 1; INK
5;"SRRRSRRSRRRRSRSRSRSRRRSSRSRR"
5060 PRINT AT 4,6; INK 0;"GH";AT
 5,6;"IJK"
5070 PRINT PAPER 4; INK 7; BRIGH
T 1;AT 18,1;" PUNTOS [00000] REC
ORD [00000]";AT 20,1;" NIVEL [1]
  PECES [DDDDDDDDDD]"
5999 GO SUB 900+(ni*100)
6000 REM  PeCeS
6005 FOR n=1 TO 7-ni
6010 LET l=RND*8+7: LET c=RND*13
+3: IF ATTR (l,c)<>42 OR ATTR (l
,c+12)<>42 THEN GO TO 6010
6015 PRINT AT l,c; INK 3;"D";AT
l,c+12;"D": NEXT n
6020 FOR n=1 TO 3: LET p=RND*5+2
: LET q=RND*13+15: IF ATTR (p,q)
<>42 THEN LET p=p+1
6030 PRINT INK 3;AT p,q;"D": NEX
T n
7000 RETURN
9000 RESTORE 9010: FOR n=0 TO 16
7: READ a: POKE USR "a"+n,a: NEX
T n
9010 DATA 24,40,28,18,28,28,10,2
4,24,20,56,72,56,56,80,24,16,40,
56,40,108,170,56,108,224,88,252,
88,44,31,4,4
9020 DATA 0,63,64,159,191,127,63
,15,0,252,2,249,253,255,252,192,
149,255,149,16,115,22,28,31,0,0,
0,0,192,184,150,61
9030 DATA 59,52,124,107,209,249,
207,209,201,71,45,227,19,23,251,
23,128,252,150,47,255,47,47,255
9040 DATA 0,0,56,124,84,108,56,1
24,122,122,118,172,61,39,161,114
9050 DATA 94,94,46,53,188,228,13
3,78,0,0,56,124,124,124,56,124
9060 DATA 190,189,191,188,36,44,
36,48,190,253,61,61,36,52,36,12
9070 DATA 0,0,0,64,162,213,235,1
91,0,4,42,119,174,127,237,247,0,
0,64,64,32,32,64,128,0,0,0,8,16,
32,64,128
9080 LET in=1
9099 REM  VaRiABLeS
                    i=3: LET ni=
1
9180 FOR n=6 TO 1 STEP -1: BEEP
.05,-20: PRINT #1;AT 1,0; BRIGHT
 1; PAPER 7; INK n;"  PULSA ENT
ER PARA COMENZAR   ": BEEP .01,
-10: PAUSE 5: NEXT n
9300 LET pe=0: LET yy=4: LET xx=
9: LET y=yy: LET x=xx: LET aa=5:
 LET bb=27: LET a=aa: LET b=bb:
LET a$="L": LET b$="N": LET c$="
A": LET d$="EF": LET p$="D"
9303 IF NOT in THEN RETURN
9305 IF INKEY$=CHR$ 13 THEN GO S
UB 4000: GO TO 30
9310 GO TO 9305
9399 REM  MuSiCa 3
9400 RESTORE 9410: FOR n=30 TO 5
0: BEEP .001,n: NEXT n: FOR n=50
 TO 20 STEP -1: BEEP .001,n: NEX
T n: PAUSE 30: FOR n=0 TO 23: RE
AD a,b: BEEP a,b: NEXT n
9410 DATA .17,9,.17,9,.25,9,.17,
9,.17,9,.25,9,.17,9,.17,12,.17,5
,.17,7,.4,9,.17,10,.17,10,.17,10
,.12,10,.17,10,.17,9,.25,9,.12,9
,.17,12,.17,12,.35,10,.8,7,1.5,1
,5
9420 FOR n=30 TO 0 STEP -.5: BEE
P .01,n: NEXT n
9: LET y=yy: LET x=xx: LET aa=5:
 LET bb=27: LET a=aa: LET b=bb:
LET a$="L": LET b$="N": LET c$="
A": LET d$="EF": LET p$="D"
9303 IF NOT in THEN RETURN
9305 IF INKEY$=CHR$ 13 THEN GO S
UB 4000: GO TO 30
9310 GO TO 9305
9399 REM  MuSiCa 3
9400 RESTORE 9410: FOR n=30 TO 5
0: BEEP .001,n: NEXT n: FOR n=50
 TO 20 STEP -1: BEEP .001,n: NEX
T n: PAUSE 30: FOR n=0 TO 23: RE
AD a,b: BEEP a,b: NEXT n
9410 DATA .17,9,.17,9,.25,9,.17,
9,.17,9,.25,9,.17,9,.17,12,.17,5
,.17,7,.4,9,.17,10,.17,10,.17,10
,.12,10,.17,10,.17,9,.25,9,.12,9
,.17,12,.17,12,.35,10,.8,7,1.5,1
,5
9420 FOR n=30 TO 0 STEP -.5: BEE
P .01,n: NEXT n
9429 REM  PaNTaLLa FiNaL
9430 CLS : INK 0: PLOT 7,7: DRAW
 PAPER 3;241,0: DRAW PAPER 3;0,1
29: DRAW PAPER 3;-241,0: DRAW PA
PER 3;0,-129
9440 PRINT AT 0,0; INK 7; PAPER
3;"  MUY BIEN ! PARA TI SE ACABA
RON EL FRIO Y LA NIEVE PARA SIEM
PRE YA QUE TE HAS GANADO  UN CRU
CE- POR LAS ISLAS DEL CARIBE.
 "
9450 PLOT 247,55: DRAW INK 1;-11
5,0: INK 6: DRAW 35,-47,2: PLOT
135,55: DRAW -95,-13,.3
9455 INK 4: PLOT 135,56: FOR n=0
 TO 7: DRAW -13+(RND*5),9-n,2: N
EXT n
9470 INK 2: PLOT 22,15: DRAW 0,7
5,.3: PLOT 30,15: DRAW 0,75,.2:
FOR n=16 TO 91 STEP 1.4: PLOT 24
,n: DRAW 5,0: NEXT n
9480 INK 4: PLOT 30,90: DRAW 17,
-30,-1.5: DRAW -15,35,2: DRAW 30
,-31,-1.5: DRAW -30,37,2: DRAW 2
1,-3,-1.5: DRAW -23,7,2
9490 PLOT 29,107: DRAW -3,7,2: D
RAW -3,-7,2: DRAW -15,-3,2: PLOT
 8,98: DRAW 7,3,-1.5: PLOT 8,89:
 DRAW 6,4,-1.5: DRAW -2,-25,1.5:
 DRAW 10,25,-1
9495 INK 6: FOR n=1 TO 10 STEP .
3: CIRCLE 200,100,n: NEXT n: INK
 2: FOR n=45 TO 105 STEP 1.6: PL
OT n,29: DRAW 7,-17: NEXT n
9496 PRINT AT 17,8; INK 7;"█": P
LOT 70,40: DRAW 2,5: PRINT AT 17
,25; INK 2;"L"
9497 PRINT AT 17,26; INK 2; PAPE
R 8;"T": PAUSE 4: PRINT AT 17,26
; INK 2; PAPER 8;"U"
9499 REM  FiN dE JUeGo
9500 LET re=2: LET pu=9: IF re<p
u THEN LET re=pu
9505 PRINT #1;AT 1,0; BRIGHT 1;
PAPER 7; INK 1;" QUIERES VOLVER
A JUGAR? (S/N)  "
9510 IF INKEY$="s" THEN GO TO 90
80
9520 IF INKEY$="n" THEN GO TO 99
00
9530 BEEP .08,-37: BEEP .05,-37:
 GO TO 9497
9900 BORDER 7: PAPER 7: INK 0: C
LS
9910 PRINT AT 0,0;"  10>REM Puls
a ENTER y adio";CHR$ 8; OVER 1;"
"";"s";#1;"NEW "; FLASH 1;"L"
9920 IF INKEY$<>CHR$ 13 THEN GO
TO 9920
9930 PRINT USR 0
```

SI BUSCAS LO MEJOR

LOS PRIME

DEL



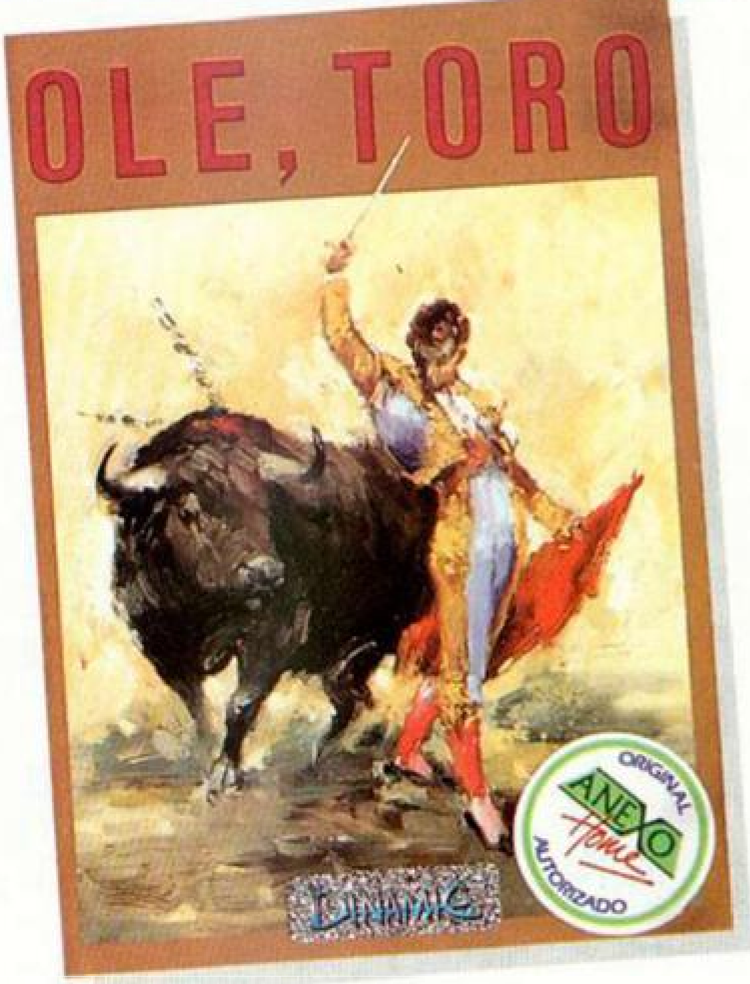


N.O.M.A.D.
Especificaciones del N.O.M.A.D. 471: Droide autónomo de combate. Alta inteligencia. Receptores visuales por infrarrojos. Reactores funcionales de avance y retroceso de gran maniobrabilidad. Alta velocidad. Cuerpo recubierto de titanio. Dos Magnum calibre 57. Esta es su primera y probable mortal misión. Las oportunidades de sobrevivir son escasas... ¡¡La libertad del universo está en tus manos!!

MIKIE
Ligón, caradura, av
Así es Mikie, el pe
controlar y que lo m
cocinas del colegio
de las chicas. El p
en Inglaterra.

OLE, TORO • Arcade • Dinamic

CITA A LAS CINCO



Hoy, a las cinco de la tarde, bajo el permiso de la presidencia y si el tiempo no lo impide, serán lidiados 6 toros 6, de la acreditada ganadería de los hermanos Ruiz con divisa blanca y azul, por un prestigioso cartel de los mejores espadas.

La Real Maestranza se encuentra llena hasta la bandera. El público, entusiasta y animado, espera a ritmo de pasodoble, el comienzo de la fiesta.

Ahí tenemos ya, en el centro del ruedo, al valiente matador que se encargará de lidiar al primer toro de la tarde. Suenan los clarines, se abre el toril y aparece como un rayo, dejando tras de sí una estela de polvo, Currito: negro zaíno, astilargo, cornituerto y rabiblanco, llevando con gallardía sus 350 kg. de peso.

Currito es recibido con una media verónica que arranca los primeros aplausos del público; rápidamente el noble animal da media vuelta y arremete con furia sobre el rojo capote.

Una chicuelina, otra verónica, un pase de rodillas..., los capotazos se suceden artísticamente y el público comienza a animarse: ¡Olé!, ¡OOlé! ¡OOOOlé!. Plas, plas, plas, plas. La plaza entera puesta en pie ovaciona al triunfador, quien desde el mismo centro de la arena se quita la montera, recibe los aplausos con orgullo y saluda.

Así es como se desarrollan las cosas en este «Olé, Toro», un programa auténticamente original cargado de una gran emoción y realismo.

Este juego, como ya sabréis si leéis asiduamente nuestra revista, ha sido realizado con el asesoramiento de expertos en temas taurinos, los cuales han colaborado tanto en el diseño de los gráficos que configuran el escenario (la Real Maestranza de Sevilla), como en los movimientos del torero y en la composición de los temas musicales que acompañan y ambientan el juego.











Así, a primera vista, todo parece excelente: la idea original, los gráficos de calidad y el sonido brillante. Pero a la hora de la verdad, este «Olé, Toro» decepciona un poco.

Con todos estos condicionantes a su favor, Dinamic podía haber realizado un programa mucho más atractivo. ¿Y qué es entonces lo que falla?, pues el tema de siempre. Que no queda lugar en el juego para la imaginación del jugador.

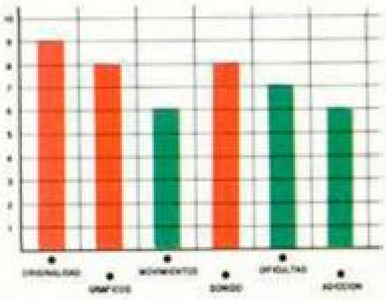
Se echa totalmente en falta el que nosotros seamos los que realmente controlemos al torero en lugar de tener que limitarnos a apretar una tecla cada vez que el toro se nos viene encima. Una vez pulsada, automáticamente se ejecuta el pase sin que nosotros podamos elegirlo ni llevarlo a cabo y lo único que tenemos que hacer es calcular la distancia adecuada, bien para que el matador dé el pase, para que el picador lance su puya, para que el banderillero se precipite hacia el toro o bien para que se realice una certera estocada. Así, todo. Prácticamente el desarrollo del juego se controla con apretar una tecla.

Comentarios a parte, el juego consiste en lo siguiente. Debemos ir superando una a una las distintas suertes en las que se divide una faena, es decir: capote, picador o suerte de varas (para decirlo a lo fino), banderillas, muleta y matar.

Para poder acceder a la siguiente se debe realizar entre las distintas oportunidades con las que contamos, una media de cinco puntos como mínimo, para lo cual se nos otorgará una puntuación de cero a diez en cada una de las acciones que realicemos, según (en teoría) el estilo y arte con los que las llevemos a cabo, pero que simplemente corresponden a que lo hagamos en el momento justo o no.

Así, aunque no hayamos sufrido ninguna cojida, si no llegamos a cinco en cada suerte no podremos pasar a la siguiente. Una vez hayamos dado muerte a un toro podremos volver a intentarlo con otro (hasta un total de seis), pero que tendrá otras características propias para ser bien toreado, (es decir, que variará la distancia).

En definitiva, «Olé, Toro» es un juego que debido a su gran originalidad podía haber dado mucho más de sí convirtiéndose en una verdadera estrella, pero que se ha quedado, tan sólo, en un programa más que, a pesar de todo, seguro que alcanzará un gran éxito por el tema que trata.



CONCURSO «OLE, TORO»

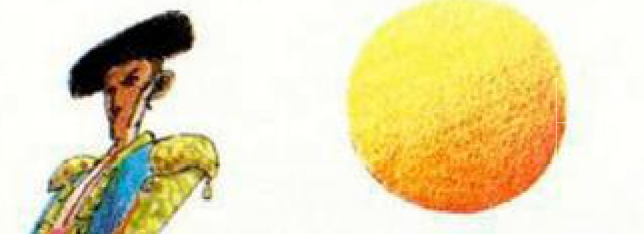
Tal y como sucedía en el programa anterior de Dinamic, West Bank, cuando llegas a un determinado momento de «Olé, Toro» aparece en la pantalla un mensaje haciendo referencia a tus habilidades como matador.

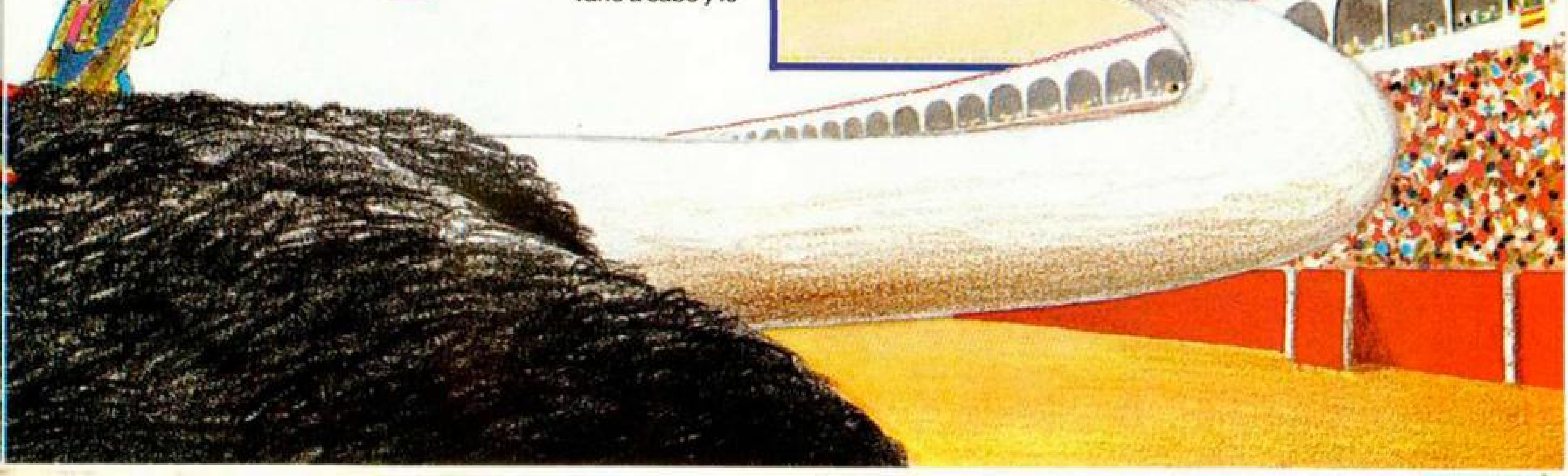
La pregunta surge por sí sola. ¿Qué dice esta frase?, si conoces la respuesta envíanos rápidamente un sobre que contenga la solución y que especifique claramente tu nombre y dirección.

Entre las cartas recibidas sortearemos diez estupendas camisetas decoradas con el bonito dibujo de la portada de Rocky.

Anímate, envía lo antes posible tu respuesta y ¡suerte!







SEAS OF BLOOD • Arcade • Adventure Internantional

NIDO DE LADRONES

Adventure International, una de las casas de software más prestigiosas en lo que a aventuras se refiere, ha lanzado un nuevo programa para el deleite de los amantes de este tipo de juegos. Se trata de Seas of Blood (Mares de sangre), que en esta ocasión nos transporta, a bordo de un legendario barco pirata, a través del intrincado mar de Inland.

La ciudad de Tak, situada al Norte de dicho mar, es el mayor nido de ladrones, piratas y gente del mal vivir que jamás se haya visto en mundo civilizado. En este lugar se encuentran los mayores asesinos, los peores criminales y los cerdos más despiadados. Uno de esos villanos eres tú.

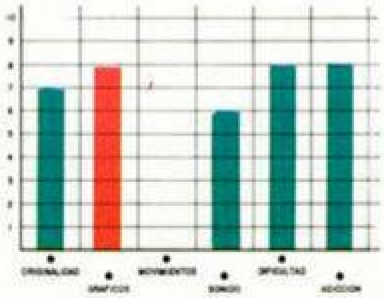
Eres el capitán del bajel pirata Banshee y junto a tu tripulación de forajidos deberás buscar denodadamente una fortuna. Para conpletar el viaje tendrás que encontrar y recoger veinte tesoros y llevarlos hasta lo alto de la montaña de Nippur, situada en el extremo Sur del mar de Inland.

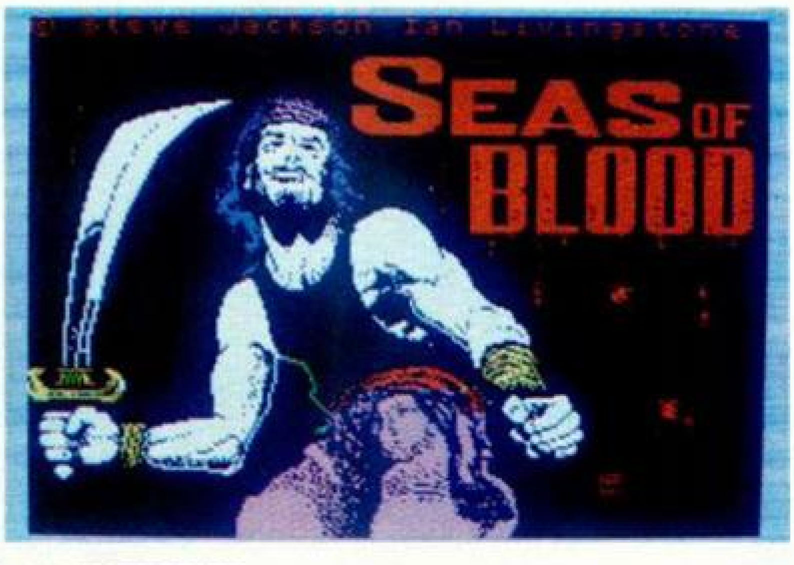
El desarrollo del juego es idéntico al de cualquier otro programa conversacional, es decir, mediante el diálogo con el ordenador que nos irá indicando en todo momento dónde nos encontramos, qué es lo que hay en nuestro alrededor y qué es lo que llevamos encima. Pero este Seas of Blood tiene un detalle bastante curioso que le da un toque original.

Se trata de que cuando decides librar un combate con algún navío al que intentas saquear, aparecen en la pantalla unos dados. En cada batalla tanto tú como tu enemigo deberéis lanzarlos y aquél que obtenga una mayor puntuación será el vencedor. Un detalle sin demasiada importancia pero que hará que además de inteligencia y astucia, necesites que la suerte esté de tu lado para lograr tus objetivos.

En cuanto a la presentación y gráficos del juego: excelente, como todas las aventuras de Adventure International.







CHIKIN CHASE • Arcade • Firebird

EL REY DEL GALLINERO

Otro de los juegos que componene la serie de Firebird llamada Super Silver es este Chikin Chase, que como su propio nombre indica mismamente en inglés, trata sobre la persecución de un pollo. Bueno, el pollo no es el perseguido, sino el perseguidor, es decir que no hay nadie que intente capturarle para darse una suculenta cena, sino que será él quien tendrá que evitar que otros se den un festín a su costa. Y decimos a su costa porque algunos malintencionados roedores intentarán comerse nada más y nada menos que a sus propios hijos, léase huevos, de sus entrañas y su corazón. Y claro, aunque ya sabemos que a estos casanovas de gallinero no les preocupan en exceso estos temas (total, diez o quince hijos de menos tampoco se van a notar mucho) a quien sí les importa es a sus numerosas «esposas», quienes aprovecharán el menor descuido de éste para asertarle un enérgico y contundente garrotazo en plena cresta.

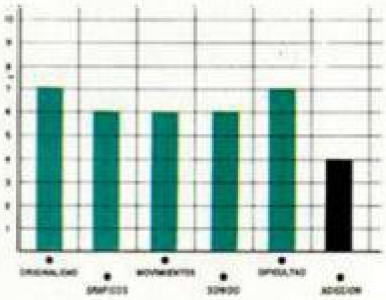




Así que ya sabe, si quiere tener contentas a sus gallinitas, tendrá que cuidar con mucho esmero de que nadie consiga llevarse ni un sólo huevo del gallinero, proeza ésta que sólo podrá relizar con suma habilidad y mucho grano, pues deberá además aprovechar los escasos momentos de relax para recuperar las valiosas energías que seguramente perderá en sus encuentros con ratones, comadrejas y demás alimañas sin encrúpulos.

En cuanto a los gráficos y todas esas cosas, pues nada especial. No están mal del todo, son graciosos y bonitos, pero nada más. Lo dibujos son monos, están bien, pero normalitos. El movimiento está bien realizado, resulta simpático, pero no aporta nada nuevo.

Así que, para resumir, un programa del montón, pero del montón bajo además.





COMANDO • Arcade • Elite

LA ACCION COMO PROTAGONISTA

La historia de este programa es realmente particular. Todo comenzó con la versión original que se realizó para las máquinas de video de los billares y ésta alcanzó un éxito tal, sobre todo en los Estado Unidos, que práticamente no quedó uno de estos locales sin poseer uno de estos juegos. Por esta razón se llevó a cabo una versión para ordenador de la cual la compañía de software británica Elite adquirió todos los derechos. Pero las cosas no iban a resultar nada sencillas. Pronto surgirían problemas debido a que otra compañía japonesa realizó un programa llamado «Who dares wins» que según Elite y Capcon, la casa creadora del Comando, infrigía estos derechos de copyright, lo que motivó que el asunto se llevara a los tribunales y se prohibiera la distribución de éste juego por una temporada. Todo esto, evidentemente, iba a despertar en el público un cierto interés por el programa que se vio incrementado aún más con la realización de una película basada en su argumento. Era la primera vez que se realizaba una cosa así y la espectación que produjo fue enorme.





Pero por fin, tras todas estas circunstancias hizo su aparición en el mercado el esperado y anhelado Comando. Cuando la primera copia de este programa llegó a nuestra redacción, la verdad es que fue acogido con mucho revuelo y ráudos corrimos a cargarlo en uno de nuestros Spectrum.

La respuesta fue unánime: —¿Y para esto tanto?

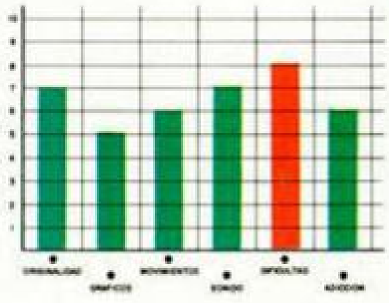
Hemos de confesar que nos llevamos una gran desilusión.

No por el tema del juego, que ya conocíamos por las máquinas tragaperras y por las versiones para otros ordenadores, sino por la mala calidad de sus gráficos.

Quizás si Comando no hubiera aparecido con esos aires de estrella y hubiera venido como un programa más, nos habría incluso gustado, pero al esperar algo especial nos pareció hasta malo.

De todas formas, como suponemos que muchos de vosotros habréis adquirido una copia de Comando, quizás este comentario llegue un poco tarde, pero para aquéllos que aún no lo hayáis hecho, os advertimos que aunque este juego está bastante bien y en general es divertido y emocionante, manteniéndote durante todo el desarrollo en una gran tensión, a nosotros personalmente no nos parece ninguna maravilla.





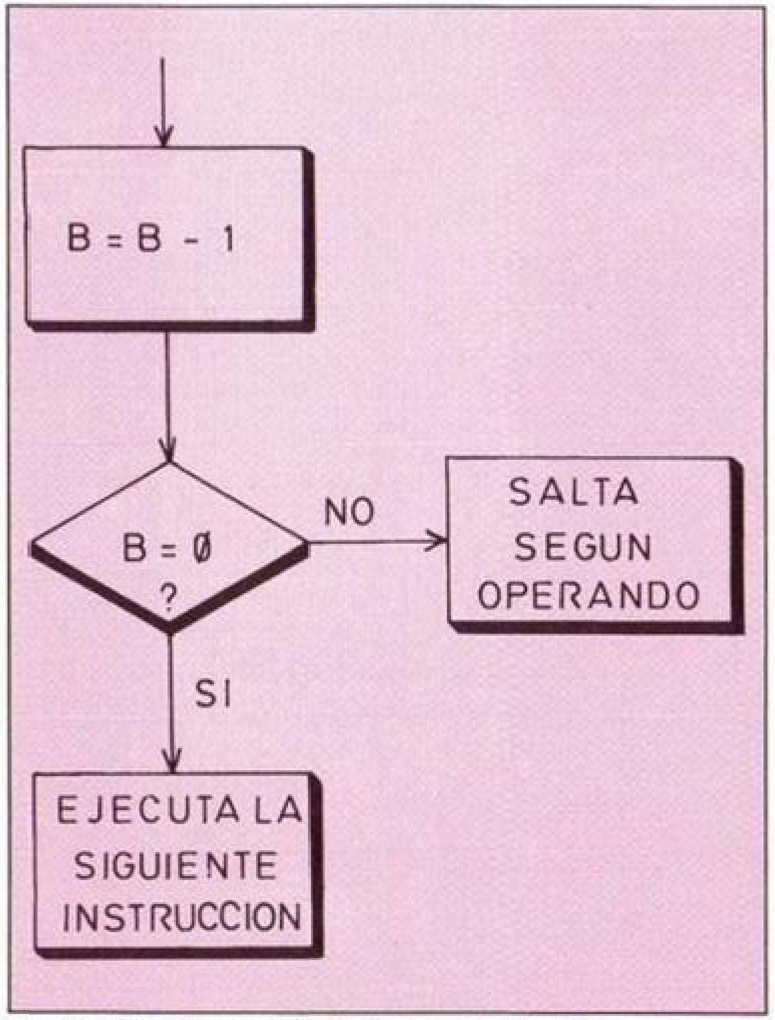


Fig. 7-2. Organigrama de la instrucción "DJNZe»

¿Qué ocurre cuando el multiplicador es igual a "0"?; dado el funcionamiento de "DJNZ", multiplicaríamos por 256. Esto provocaría un resultado erróneo, de forma que tendremos que detectar cuando el multiplicador sea "0" para actuar en consecuencia.

Por otro lado, es necesario tomar los valores iniciales de algún sitio y almacenarlos en algún otro sitio. Colocaremos nuestra rutina en el buffer de impresora (aunque es totalmente reubicable) y utilizaremos las primeras direcciones como variables donde almacenar los datos de entrada y el resultado; "VAR1" en una variable de 2 octetos en las direcciones 23296 y 23297, en la entrada contiene el multiplicando y en la salida, los dos octetos menos significativos del resultado; "VAR2" es una variable de 1 octeto en la dirección 23298, en la entrada contiene el multiplicador y en la salida, el octeto más significativo del resultado. Dejamos otro byte libre y colocamos la rutina a partir de 23300 para que empiece en un número redondo que es más fácil de recordar. Hemos llamado a la rutina "MULTI" y su listado completo es:

```
130 MULTI LD   HL,VAR2
140       LD   B,(HL)
150       LD   DE,(VAR1)
160       DEC  B
170       INC  B
180       JR   Z,CERO
190       XOR  A
200       LD   HL,0
210 BUCLE ADD  HL,DE
220       JR   C,CARRY
230 LOOP  DJNZ BUCLE
240       LD   (VAR1),HL
250       LD   (VAR2),A
260       RET
270 CARRY INC  A
280       JR   LOOP
290 CERO  LD   HL,0
300       LD   (VAR1),HL
310       RET
```

Las líneas 130 y 140 cargan en "B" el contenido de "VAR2"; en 150 se carga en "DE" el contenido de "VAR1"; las líneas 160, 170 y 180 comprueban si "B" es "0", en cuyo caso, se salta a la línea 290. En 190 y 200 se carga "0" en "A" y "HL"; 210, 220 y 230 constituyen el bucle principal; si se produce acarreo, se incrementa "A" en 270 y 280. Finalmente, las líneas 240 y 250 cargan en "VAR1" y "VAR2" el resultado antes de retornar.

En la FIGURA 7-5 se puede ver el organigrama de esta rutina; un estudio detenido de este organigrama contribuirá a una mejor comprensión de su funcionamiento.

Vamos a ensamblarla. Las líneas 130 y 170 no deben dar problemas, simplemente, recuerde sustituir "VAR1" y "VAR2" por sus respectivos

Contenido del registro PC al leer la instrucción la CPU

(PC):	10000000	80h
	00100101	25h

Indicador C=0

Instrucción

JR C,+10:	00111000	38h
	00001000	08h

(Ensamblamos 08h que es 10–2, es decir, e–2).

Contenido del registro PC después de la ejecución

(PC):	10000000	80h
	00100101	25h

No se cumplía la condición porque "C" estaba a "0", de forma que no se verifica el salto y pasa a ejecutarse la instrucción que sigue en el orden normal del programa.

JR NC,e

OBJETO:

Si el indicador de acarreo "C" no está activo; salta a la posición de memoria que resulta de añadir a la propia posición de la instrucción el entero de desplazamiento "e", el cual puede adquirir los valores desde –126 a +129.

Si el indicador de acarreo "C" está activo, ejecuta la siguiente instrucción.

CODIGO DE MAQUINA:

00110000	30h
<——e–2——>	

INDICADORES DE CONDICION QUE AFECTA:

Ninguno

CICLOS DE MEMORIA:

Si cumple la condición, 3

Si no cumple la condición, 2

CICLOS DE RELOJ:

Si cumple la condición, 12

Si no cumple la condición, 7

EJEMPLO:

En la dirección absoluta 7743h

JR NC,–7

Contenido del registro PC al leer la instrucción la CPU

(PC):	01110111	77h
	01000101	45h

Indicador C=0

Instrucción

JR NC,–7:	00110000	30h
	11111011	FBh

Contenido del registro PC después de la ejecución

(PC):	01110111	77h
	00111110	3Eh

La siguiente instrucción a ejecutar será la de la posición de memoria 773Eh; por tanto, se ha efectuado el salto.

JR Z,e

OBJETO:

Si el indicador de cero "Z" está activo; salta a la posición de memoria que resulta de añadir a la propia posición de la instrucción el entero de desplazamiento "e", el cual puede adquirir los valores desde –126 a +129.

Si el indicador de cero Z no está activo; ejecuta la siguiente instrucción.

CODIGO DE MAQUINA:

00101000	28h
<——e–2——>	

INDICADORES DE CONDICION QUE AFECTA:

Ninguno

CICLOS DE MEMORIA:

Si cumple la condición, 3

Si no cumple la condición, 2

CICLOS DE RELOJ:

Si cumple la condición, 12

Si no cumple la condición, 7

EJEMPLO:

En la dirección absoluta 4590h

JR Z,+6

Contenido del registro PC al leer la instrucción la CPU

(PC):	01000101	45h
	10010010	92h

Indicador Z=1

Instrucción

```
JR Z,+6:  | 00101000 | 28h
          | 00000100 | 04h
```

Contenido del registro PC después de la ejecución

```
(PC):  | 01000101 | 45h
       | 10010110 | 96h
```

La siguiente instrucción a ejecutar será la de la posición de memoria 4596h; se efectúa el salto.

JR NZ,e

OBJETO:

Si el indicador de acarreo "Z" no está activo; salta a la posición de memoria que resulta de añadir a la propia posición de la instrucción el entero de desplazamiento "e", el cual puede adquirir los valores desde −126 a +129.

Si el indicador de acarreo "Z" está activo; ejecuta la siguiente instrucción.

CODIGO DE MAQUINA:

```
| 00100000 | 20h
| <--e-2--> |
```

INDICADORES DE CONDICION QUE AFECTA:

Ninguno

CICLOS DE MEMORIA:

Si cumple la condición,
3

Si no cumple la condición,
2

CICLOS DE RELOJ:

Si cumple la condición,
12

Si no cumple la condición,
7

EJEMPLO:

En la dirección absoluta 5000h

```
JR NZ,-126
```

Contenido del registro PC al leer la instrucción la CPU

```
(PC):  | 01010000 | 50h
       | 00000000 | 00h
```

Indicador Z=1

Instrucción

```
JR NZ,-126:  | 00100000 | 20h
             | 10000000 | 80h
```

Contenido del registro PC después de la ejecución

```
(PC):  | 01010000 | 50h
       | 00000010 | 02h
```

Ejecuta la siguiente instrucción y no se verifica el salto.

-.-.-.-.-.-.-

Existe, dentro del repertorio del Z-80, una instrucción compleja que resulta especialmente adecuada para establecer bubles de iteración; esta instrucción es "DJNZ", abreviatura de: "Decrement and Jump if Not Zero", en castellano: "Decrementa y Salta si No es Cero". Esta instrucción puede considerarse equivalente a la sentencia "FOR...NEXT" del Basic, ya que se usa para lo mismo, si bien, de distinta forma.

DJNZ,e

OBJETO:

Decrementa el registro "B" (le resta "1").

Si el valor del registro "B" es distinto de cero; salta a la posición de memoria que resulta de añadir a la propia posición de la instrucción el entero de desplazamiento "e", el cual puede adquirir los valores desde −126 a +129.

Si el valor del registro "B" es cero; ejecuta la siguiente instrucción (ver FIGURA 7-2).

CODIGO DE MAQUINA:

```
| 00010000 | 10h
| <--e-2--> |
```

INDICADORES DE CONDICION QUE AFECTA:

Ninguno

CICLOS DE MEMORIA:

Si registro "B" diferente de cero,
3

Si registro "B" igual a cero,
2

CICLOS DE RELOJ:

Si registro "B" diferente de cero,
13

Si registro "B" igual a cero,
8

EJEMPLO:

En la dirección absoluta 4723h

```
DJNZ,-20
```

Contenido del registro "B"

Ejemplos

Los ejemplos de este capítulo van a ser bastante más complicados que los de capítulos anteriores; el poder utilizar ya las instrucciones de salto, nos va a permitir crear bucles y hacer cosas más útiles. Desarrollaremos una rutina para multiplicar, otra para dividir y, finalmente, una rutina que borra la pantalla por trozos, dependiendo el trozo borrado del valor que contenga el acumulador.

Vamos a empezar por la rutina de multiplicar. Algunas CPUs de ordenadores mayores que el Spectrum, incluyen la instrucción de multiplicar en su Assembler; como no es el caso del Z-80, tal vez resulte útil disponer de una rutina de multiplicación que se podrá incluir dentro de un programa en cualquier momento que se necesite.

En el producto AxB, vamos a llamar "multiplicando" a "A" y "multiplicador" a "B". Básicamente, multiplicar "AxB" consiste en sumar "A" sobre sí mismo tantas veces como indique "B"; parece un trabajo bastante adecuado para un bucle. Supongamos que tenemos el multiplicando en el registro "DE" y el multiplicador en el registro "B"; el bucle podría ser:

```
        LD    HL,0
BUCLE   ADD   HL,DE
        DJNZ  BUCLE
```

El contenido del registro "DE" se iría sumando sobre "HL" tantas veces como indicara el contenido "B". Evidentemente, el multiplicando no puede ser superior a 65535 (FFFFh) ni el multiplicador puede ser superior a 255 (FFh). Pero si multiplicamos 65535 por 255 obtenemos 16711425 (FEFF01h) y este número no se puede representar con dos octetos, de forma que, a lo largo de la ejecución del bucle, obtendremos varias veces un acarreo en el registro "HL"; si no queremos que el resultado no tenga nada que ver con la realidad, será mejor que llevemos la cuenta de las veces que se produce acarreo. Algo más correcto sería:

```
        XOR   A
        LD    HL,0
BUCLE   ADD   HL,DE
        JR    C,CARRY
```

"cc"	Código	Significado	Indicador
NZ	000	no cero	Z=0
Z	001	cero	Z=1
NC	010	no acarreo	C=0
C	011	acarreo	C=1
PO	100	paridad impar o no desbordamiento	P/V=0
PE	101	paridad par o desbordamiento	P/V=1
P	110	signo positivo	S=0
M	111	signo negativo	S=1

Fig. 7-1. Tabla de condiciones "cc" para la instrucción de salto "JP".

```
LOOP    DJNZ  BUCLE
        .....
        .....
        RET
CARRY   INC   A
        JR    LOOP
```

Los puntos suspensivos indican que ahí tendrán que ir algunas instrucciones que hagan algo con el resultado antes de retornar. La diferencia entre esta rutina y la otra, es que detectamos cada vez que hay un acarreo y lo acumulamos en "A"; de esta forma, el resultado final de nuestra multiplicación vendrá dado por los contenidos de "A" y "HL" puestos en el orden: "AHL", es decir, "A" será el octeto de mayor peso y "L" será el de menor.

Persiste aún un problema:

OBJETO:

Salta a la posición de memoria direccionada por el contenido del registro índice IX.

CODIGO DE MAQUINA:

11011101	DDh
11101001	E9h

INDICADORES DE CONDICION QUE AFECTA:

Ninguno

CICLOS DE MEMORIA:

2

CICLOS DE RELOJ:

8

EJEMPLO:

JP (IX)

Contenido del registro índice IX.

(IX):	10110000	B0h
	11110000	F0h

Instrucción

JP (IX):	11011101	DDh
	11101001	E9h

Contenido del registro "PC" después de la ejecución.

(PC):	10110000	B0h
	11110000	F0h

La siguiente instrucción a ejecutar será la de la posición de memoria B0F0h.

JP (IY)

OBJETO:

Salta a la posición de memoria direccionada por el contenido del registro índice IY.

CODIGO DE MAQUINA:

11111101	FDh
11101001	E9h

INDICADORES DE CONDICION QUE AFECTA:

Ninguno

CICLOS DE MEMORIA:

2

CICLOS DE RELOJ:

8

EJEMPLO:

JP (IY)

Contenido del registro índice IY

(IY):	01110110	76h
	10000100	84h

Instrucción

JP (IY):	11111101	FDh
	11101001	E9h

Contenido del registro "PC" después de la ejecución.

(PC):	01110110	76h
	10000100	84h

La siguiente instrucción a ejecutar será la de la posición de memoria 7684h.

-.-.-.-

Las instrucciones "JP (IX)" y "JP (IY)" son de escasa utilidad, ya que los registros índices rara vez se utilizan para direccionar saltos. Estas instrucciones son, más bien, una consecuencia del procedimiento que utiliza el microprocesador para decodificar las instrucciones que llevan direccionamiento indexado.

Si mira atentamente las tablas de codificación, verá que cualquier instrucción que utilice direccionamiento indirecto a través del registro "HL", puede funcionar con direccionamiento indexado si **se antepone al código de operación "DD" para el índice "IX" o "FD" para el índice "IY"**; esto es una regla general.

Hubiera sido más útil disponer de las instrucciones "JP (BC)" y "JP (DE)"; por desgracia, el repertorio del Z-80 no incluye estas instrucciones; no obstante, es posible simularlas mediante un artificio que utiliza en varias ocasiones el Sistema Operativo. El procedimiento se comprenderá mejor cuando estudiemos las subrutinas, pero podemos anticipar que consiste en "engañar" al microprocesador metiendo en la pila el contenido del registro y haciendo "RET" (retorno desde subrutina), con lo que tomará el valor que hemos metido en la pila como dirección de retorno. Por ejemplo, la instrucción "JP (BC)" se puede simular con: "PUSH BC" + "RET". Esta es una de las ventajas de compartir la pila de usuario con la máquina.

-.-.-.-.-

(B):	00000101	05h

Contenido del registro PC al leer la instrucción la CPU

(PC):	01000111	47h
	00100101	25h

Instrucción

DJNZ,-20h:	00010000	10h
	11011110	DEh

Contenido del registro "B" después de la ejecución

(B):	00000101	04h

Contenido del registro PC después de la ejecución

(PC):	01000111	47h
	00000011	03h

La siguiente instrucción a ejecutar será la de la posición de memoria 4703h. Se ha decrementado el registro "B", pero como todavía no ha llegado a "0", se efectúa el salto.

Si el registro "B" hubiera contenido "1" antes de la ejecución, al decrementarlo hubiera pasado a valer "0" con lo que no se habría producido el salto, y se hubiera ejecutado la siguiente instrucción.

A continuación, vamos a **ver cómo se utilizan ésta y** otras instrucciones para crear, en código máquina, bucles de diferentes tipos.

Bucles

Por fin, ha llegado el momento de estudiar la que es, sin duda, la principal técnica de programación. En código máquina (o en Assembler), necesitaremos uno o más **bucles para casi cualquier cosa** que queramos hacer, así que recomendamos al lector que ponga mucha atención en esta parte, y la lea las veces que sean necesarias hasta que consiga comprenderlo perfectamente.

Al igual que en Basic, existen varias formas de hacer un bucle; la más sencilla consiste en escribir una serie de instrucciones y terminar con un "GO TO" que mande, de nuevo, a la primera de ellas. En Basic sería algo así:

```
10 REM Comienza el bucle
20 ......
30 ......
40 ......
50 GO TO 10
```

Este bucle tiene un fallo gravísimo: el ordenador se queda atrapado en él eternamente; claro que esto en Basic no es problema ya que siempre podemos hacer "BREAK"; pero en Assembler no vamos a disponer de un "BREAK" tan fácilmente, así que será mejor que tengamos mucho cuidado de no crear bucles donde el ordenador se quede atrapado. Algo más correcto sería:

```
10 REM Comienza el bucle
20 .......
30 .......
40 .......
50 IF ... THEN GO TO 10
```

Esta vez el bucle sólo se cierra si se cumple la condición que pongamos entre "IF" y "THEN", en caso contrario, se ejecuta la siguiente instrucción y se sale del bucle.

En Assembler podemos conseguir un efecto similar:

```
10 BUCLE  .... ;Comienza el bucle
20        ....
30        ....
40        ....
50        JP   cc,BUCLE
```

¿Qué hemos hecho?: tenemos una serie de instrucciones que deben repetirse mientras la condición "cc" se cumpla; ponemos esas instrucciones a partir de una etiqueta (en nuestro caso: "BUCLE") y al final colocamos una instrucción que obligue al microprocesador a saltar a la instrucción donde está la etiqueta solamente si se cumple la condición. Al igual que **antes si ésta no se cumple, se** saldrá del bucle y se continuará con el proceso normal.

Los saltos absolutos tienen el grave inconveniente de no permitir la reubicación del código en un lugar distinto de donde fueron ensamblados; por otro lado, ocupan 3 bytes en lugar de los 2 que ocupa un salto relativo; así que no **parece mala idea de utilizar saltos relativos para los bucles; siempre, claro está, que las instrucciones que componen el bucle no ocupen más de 127 bytes.**

En el ejemplo anterior, podemos cambiar la línea 50 por:

```
50        JR      cc,BUCLE
```

La condición "cc" puede ser cualquiera de: "Z", "NZ", "C" y "NC"; al ensamblar esta línea, ocupará solamente 2 bytes y, además, correrá exactamente igual en cualquier posición de memoria.

Ahora, vamos a prestar un poco de atención a la condición "cc" que nos permite cerrar el bucle; examine este programa en Basic:

```
10 LET b=12
20 REM comienza el bucle
30 ....
40 ....
50 LET b=b-1
60 IF b<>0 THEN GO TO 20
```

Es fácil comprobar que el bucle se repetirá 12 veces; en cada pasada se resta "1" a "b"; en la duodécima pasada, "b" llegará a valer "0", con lo que no se cumple la condición "b<>0" y se sale del bucle. En Assembler, esto sería algo así:

```
10         LD    B,12
20 BUCLE   ......
30         ......
40         ......
50         DEC   B
60         JR    NZ,BUCLE
```

La lógica es la misma: cargamos "12" en el registro "B" y lo decrementamos en cada pasada, cerrando el bucle mientras "B" sea distinto de "0". ¿Por qué hemos elegido el registro "B"?: las dos instrucciones "DEC B" y "JR NZ, BUCLE" ocupan, en total, 3 bytes; pero éste es el sitio adecuado para colocar la instrucción "DJNZ" de la siguiente forma:

```
50         DJNZ  BUCLE
```

Esto es lo que se denomina un "BUCLE DE ITERACION" y el registro "B" resulta especialmente idóneo para usarlo en este tipo de bucles; aunque nada nos impide utilizar otros registros; pero tenga en cuenta que "DJNZ" sólo actúa sobre el registro "B".

Observe el siguiente programa en Basic:

```
10 LET c=15
20 REM Bucle exterior
30 LET b=12
40 REM Bucle interior
50 ......
60 ......
70 LET b=b-1
80 IF b<>0 THEN GO TO 40
90 LET c=c-1
100 IF c<>0 THEN GO TO 20
```

Es equivalente a:

```
10 FOR c=0 TO 15
20 FOR b=0 TO 12
30 .......
40 .......
50 NEXT b
60 NEXT c
```

Efectivamente, se trata de dos bucles "anidados", es decir, uno dentro del otro. En Assembler también podemos hacerlo:

```
10         LD    C,15
20 BUC_1   LD    B,12
30 BUC_2   .......
40         .......
50         .......
60         DJNZ  BUC_2
70         DEC   C
80         JR    NZ,BUC_1
```

Parece que si empezamos a anidar muchos bucles uno dentro de otro, se nos acabarán pronto los registros. Bien, no es cierto, podemos utilizar la "pila":

```
10         LD    B,18
20 BUC_1   PUSH  BC
30         LD    B,15
40 BUC_2   PUSH  BC
50         LD    B,12
60 BUC_3   ......
70         ......
80         ......
90         DJNZ  BUC_3
100        POP   BC
110        DJNZ  BUC_2
120        POP   BC
130        DJNZ  BUC_1
```

Si se toma la molestia de seguir el curso al programa, verá que tenemos anidados tres bucles, cada uno dentro del otro, todos están controlados por el mismo registro y no hay posibilidad de confusión; vamos guardando los registros en la pila y recuperándolos sólo cuando es necesario.

Hasta ahora, hemos supuesto que el número de iteraciones (veces que tiene que repetirse el bucle) era menor de 256; pero ¿y si fuera mayor?; en ese caso, tendríamos que recurrir a un registro de 16 bits como contador del bucle, por ejemplo, el "BC". Existe un pequeño inconveniente, al decrementar el "BC" no resultan afectados los indicadores, pero podemos evitarlo con un pequeño truco:

```
10         LD    BC,500
20 BUCLE   .......
30         .......
40         .......
50         DEC   BC
60         LD    A,B
70         OR    C
80         JR    NZ,BUCLE
```

¿Ingenioso verdad?; por desgracia no se nos ha ocurrido a nosotros, lo hacen todos los programadores de Assembler; se trata, simplemente, de cargar en "A" el contenido de "B" y hacerle un "OR" con "C"; el resultado sólo será "0" si ambos, "B" y "C", valen "0". Tiene el ligero inconveniente de modificar el contenido de "A", pero para eso tenemos la pila:

```
10         LD    BC,500
20         PUSH  AF
30 BUCLE   POP   AF
40         .......
50         .......
60         DEC   BC
70         PUSH  AF
80         LD    A,B
90         OR    C
100        JR    NZ,BUCLE
110        POP   AF
```

La cosa se complica un poco con los PUSH y POP, pero funciona de maravilla y el contenido de "A" no se altera en absoluto. No olvide que la línea donde se define el valor inicial del contador tiene que estar antes de la etiqueta donde se inicia el bucle; un error muy típico de principiante consiste en colocar la etiqueta en la línea donde se carga el contador con el valor inicial, algo como:

```
10 BUCLE   LD    B,2
20         .......
30         .......
40         DJNZ  BUCLE
```

Terrible error; el registro "B" siempre vale "2" y el ordenador se queda "enganchado" indefinidamente en el bucle.

Por último y para terminar con el tema de los bucles, vamos a contarle un truco: si desea iterar un bucle 256 veces, no podrá cargar el número "256" en el registro "B", pero no necesita recurrir a un registro de 16 bits; bastará con que cargue el registro "B" con "0" y el bucle se iterará 256 veces. Si desea más de 65536 iteraciones (cosa bastante improbable), lo más sencillo es anidar dos bucles; el número total de iteraciones vendrá dado por el producto de las iteraciones de cada uno de los dos bucles; con 65536 iteraciones en cada uno, hasta un microprocesador relativamente rápido como el Z-80A tardará unos cuantos segundos.

-.-.-.-.-.-.-

Instrucciones de salto indirecto

JP (HL)

Salta a la posición de memoria direccionada por el contenido del par de registros "HL".

CODIGO DE MAQUINA:

11101001	E9h

INDICADORES DE CONDICION QUE AFECTA:

Ninguno

CICLOS DE MEMORIA:

1

CICLOS DE RELOJ:

4

EJEMPLO:

JP (HL)

Contenido del par de registros "HL"

(H):	10001010	8Ah
(L):	00001011	0Bh

Instrucción

JP (HL):	11101001	E9h

Contenido del registro "PC" después de la ejecución.

(PC):	10001010	8Ah
	00001011	0Bh

La siguiente instrucción a ejecutar será la de la posición de memoria 8A0Bh. El salto es incondicional y, por tanto, se verifica siempre.

JP (IX)

LOS JUSTICIEROS DEL SOFTWARE

Made in Spain, la casa de sotfware española que se está convirtiendo en una de las más importantes del mercado, nos presentaba hace algunas semanas su última creación, «Sir Fred», que fue comentado en su momento en Microhobby. Pues bien, ahora le toca el turno a nuestros justicieros y esta es su opinión.



«Un juego imaginativo»

- **Positivo:** Es un programa hecho para los amantes de los juegos difíciles. Hay que desarrollar mucho la imaginación. Los gráficos son excelentes, tanto por la calidad como por la cantidad. El movimiento es realmente bueno. El sonido es correcto. El desarrollo del juego está muy bien pensado.
- **Negativo:** La presentación del juego contrasta con la calidad del mismo, pero se le puede perdonar perfectamente al ver que ha ganado en calidad y lujo de detalles durante el desarrollo del mismo.

Resumiendo: Un buen juego.
Puntuación: 9.



JOSE C. NUÑEZ. 18 años. Estudiante. Santander.



«Un juego difícil»

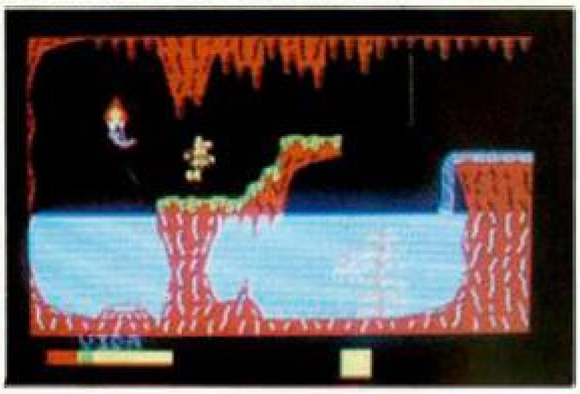
- **Positivo:** La originalidad está muy bien. Los gráficos son muy buenos y llenos de detalles de buen gusto.

Los movimientos, además de estar bien, tienes que calcular la fuerza de los mismos. El sonido está muy logrado y presente durante todo el juego.

- **Negativo:** Es un juego muy difícil, pero que con la práctica llegas a manejar bastante bien.

Resumiendo: Es un juego muy bueno y adictivo.
Puntuación: 9

DANIEL GONZALEZ. 12 años. Estudiante. Madrid.



«El movimiento no muy logrado»

- **Positivo:** Los gráficos, el color, el sonido y el personaje.
- **Negativo:** El movimiento, que salvo al andar, no está muy logrado.

Resumiendo: En general es un gran juego en el que hay que hacer numerosas acciones, con un gran nivel de dificultad, comparable a cualquier aventura, pero sin dejar de ser un buen arcade.
Puntuación: 8



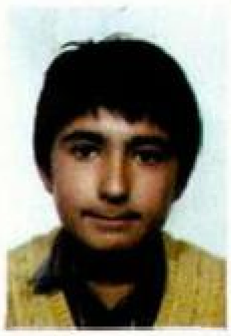
JUSTO SORIA GIL. 15 años. Estudiante. Barcelona.

Sir Fred

«Una obra maestra»

- **Positivo:** Sus gráficos y ambientación son expléndidos y reales, además de simpáticos. El movimiento supera la perfección. Lo que es muy de agradecer son los efectos sonoros que son buenos y correctos. Es un juego que crea mucha adicción. Es una obra maestra que no puede faltar en ninguna colección.
- **Negativo:** Lo único que se echa en falta es una canción al comenzar el juego, por lo demás es perfecto.

Resumiendo: La mejor video-aventura que he visto. Será número 1 muy pronto.
Puntuación: 10



GORKA POLITE. 12 años.



«Movimiento mediocre»

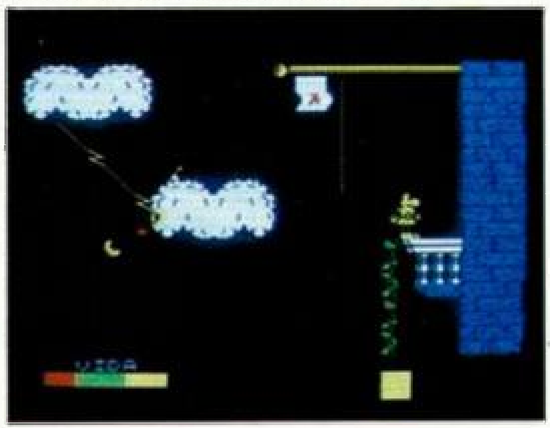
- **Positivo:** Los gráficos son bastante buenos y el movimiento es quizá lo mejor del juego. A medida que se adentra en la acción, va adquiriendo mayor interés. Los colores han sido muy bien tratados. La dificultad es muy alta y gracias a esto la adicción es mayor.

La respuesta al movimiento es buena.

- **Negativo:** El sonido es mediocre y la vida con que cuenta nuestro personaje es escasa.

Resumiendo: Es un juego muy divertido.
Puntuación: 7.

PURIFICACION HERNANDEZ. 15 años. Estudiante. Burgos.



«Buenos efectos gráficos»

- **Positivo:** Los gráficos de este juego son aceptables, destacando el simpático personaje. El movimiento es muy bueno, con una gran capacidad de respuesta. Está hecho muy bien el efecto de rodar por las escaleras, de andar, tirar con arco y luchar con la espada.

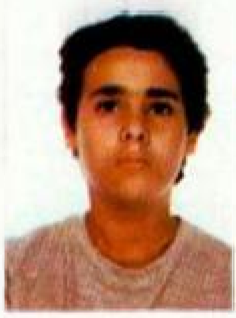
El sonido es el adecuado para un programa de este tipo.

- **Negativo:** El programa, a mi juicio, no tiene aspectos negativos.

Resumiendo: Un juego entretenido y con un nivel de dificultad muy alto.
Puntuación: 8



JAVIER REDONDO.

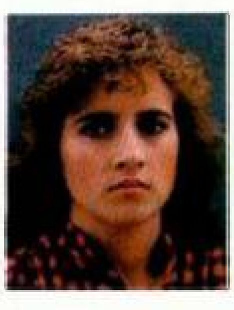


«Está lleno de detalles graciosos»

- **Positivo:** Los gráficos son muy buenos, así como el movimiento del personaje que lo hace muy real. Está lleno de detalles graciosos. El sonido es muy bueno, con unos efectos perfectos. La adicción y dificultad son muy altas.
- **Negativo:** Es difícil controlar al personaje, y la presentación no está muy conseguida.

Resumiendo: Es buen juego, difícil, pero agradable de jugar.
Puntuación: 8

JAVIER VALERO. 15 años. Estudiante. Elche (Alicante).



CELIA MENENDEZ.

«Gráficos excepcionales»

- **Positivo:** Los gráficos son excepcionales. El nivel de dificultad es medio y el colorido de las pantallas está muy bien tratado. Dentro de lo que cabe es original.
- **Negativo:** El sonido es insuficiente. El movimiento, sobre todo al andar, no está bien logrado. La pantalla de presentación es lo peor de todo.

Resumiendo: Los gráficos es lo principal, y lo que le da importancia al juego.
Puntuación: 8

LAS INTERRUPCIONES EN EL Z-80 (I)

Primitivo de FRANCISCO

El Z-80 posee un completo sistema de interrupciones que puede ser muy útil para el programador; pero éste resulta aún más interesante en aplicaciones hardware, como veremos a continuación. Las interrupciones deben ser perfectamente conocidas para la comprensión de muchos de nuestros montajes los cuales las usan como método eficaz de acceso.

El propósito de las interrupciones es permitir un perfecto acoplamiento entre la CPU (Z-80) y los periféricos. En todo proceso de interrupción es siempre el periférico el que toma la iniciativa al producirse ciertas condiciones externas predeterminadas a las cuales el periférico en cuestión es sensible.

Así, una vez generada la interrupción, el microprocesador ha de pasar a atenderla tan pronto como concluya totalmente la ejecución de la instrucción en curso en el momento de producirse la petición de interrupción (esto ha de ser así obviamente para no perder instrucciones con las interrupciones).

Una vez tomada la interrupción, el microprocesador salta a una dirección de memoria preestablecida previamente en donde el programador ha de colocar la subrutina que trata la interrupción. El microprocesador retorna al programa principal continuando a partir de la instrucción siguiente a la que estaba en proceso cuando se produjo la interrupción. El microprocesador tomará siempre como subrutina el lugar al que salte tras la interrupción, por ello anota en el stack la dirección de la instrucción siguiente que será la de retorno. Debido a esto, al final de la subrutina que trata la interrupción hay que colocar una instrucción de retorno de interrupción. En principio una vulgar instrucción de retorno serviría; pero sólo las de retorno de interrupción restablecen las condiciones iniciales en el interior del microprocesador, como se verá posteriormente.

Las interrupciones van estrechamente ligadas al hardware y nosotros las usaremos necesariamente en alguno de nuestros montajes futuros.

La filosofía de interrupciones fue incluida en los microprocesadores para dar mayor velocidad de acceso a periféricos sin necesidad de ocupar software y por ende, tiempo de proceso. Mediante este sistema, el periférico solicita ser atendido cuando éste lo precise sin necesidad de que exista en el programa un bucle que esté regularmente accediendo al periférico para averiguar si éste requiere su atención.

Un ejemplo típico es el acceso a un teclado. Sería mostruoso estar explorando constantemente toda la matriz del teclado para ver si se ha pulsado alguna tecla, esto quitaría tiempo de ejecución al programa principal. Así pues, el acceso a teclados se hace por interrupción al ser pulsada alguna tecla. En el caso del Spectrum, la ULA interrumpe cada 20 milisegundos para que el teclado sea atendido a este ritmo (recuérdese que entre otros cometidos, la ULA tiene a su cargo el hardware de acceso al teclado).

Las interrupciones en el Spectrum

Lo dicho hasta ahora responde a conceptos generales sobre interrupciones, seguidamente particularizaremos las mismas para el Z-80, pues como usuarios del Spectrum es lo que realmente nos interesa.

El Z-80 tiene dos entradas de solicitud de interrupción, denominadas: NMI (NON MASKABLE INTERRUPTION) cuyo acceso es por la pata 17 del microprocesador, e INT (MASKABLE INTERRUPT) a la cual se accede por la pata 16.



NMI (Interrupción no enmascarable)

La NMI o interrupción no enmascarable tiene PRIORIDAD ABSOLUTA, por tanto, nunca podrá ser deshabilitada por el programador, aunque éste puede ignorarla en su programa.

Este método de interrupción fue ideado para que el microprocesador pudiera atender eventos graves en un sistema, como puede ser variación en las fuentes de alimentación y otras alar-

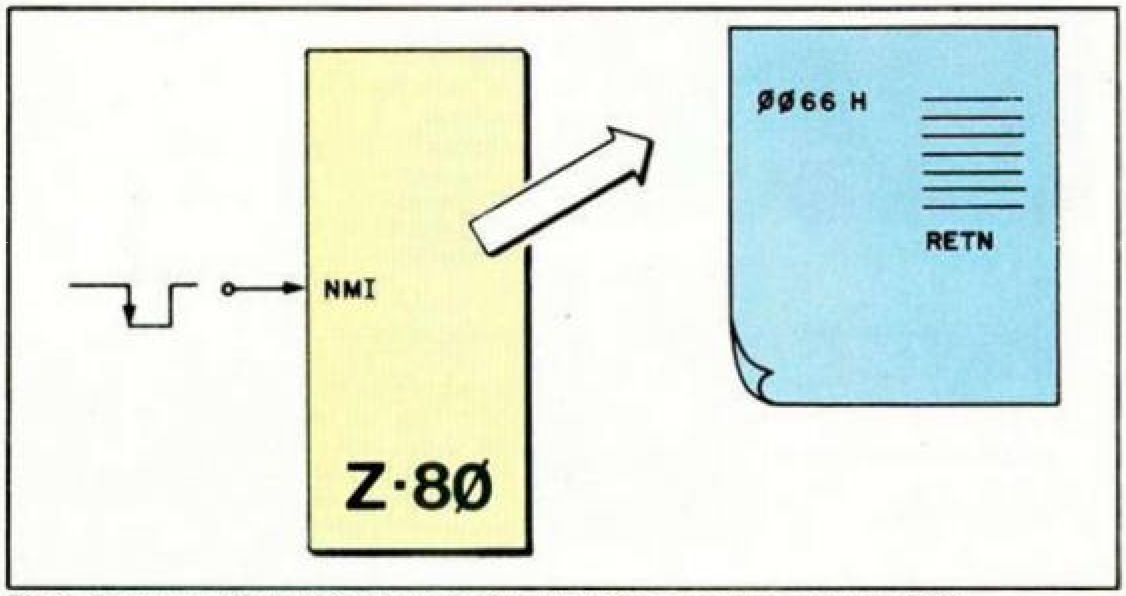
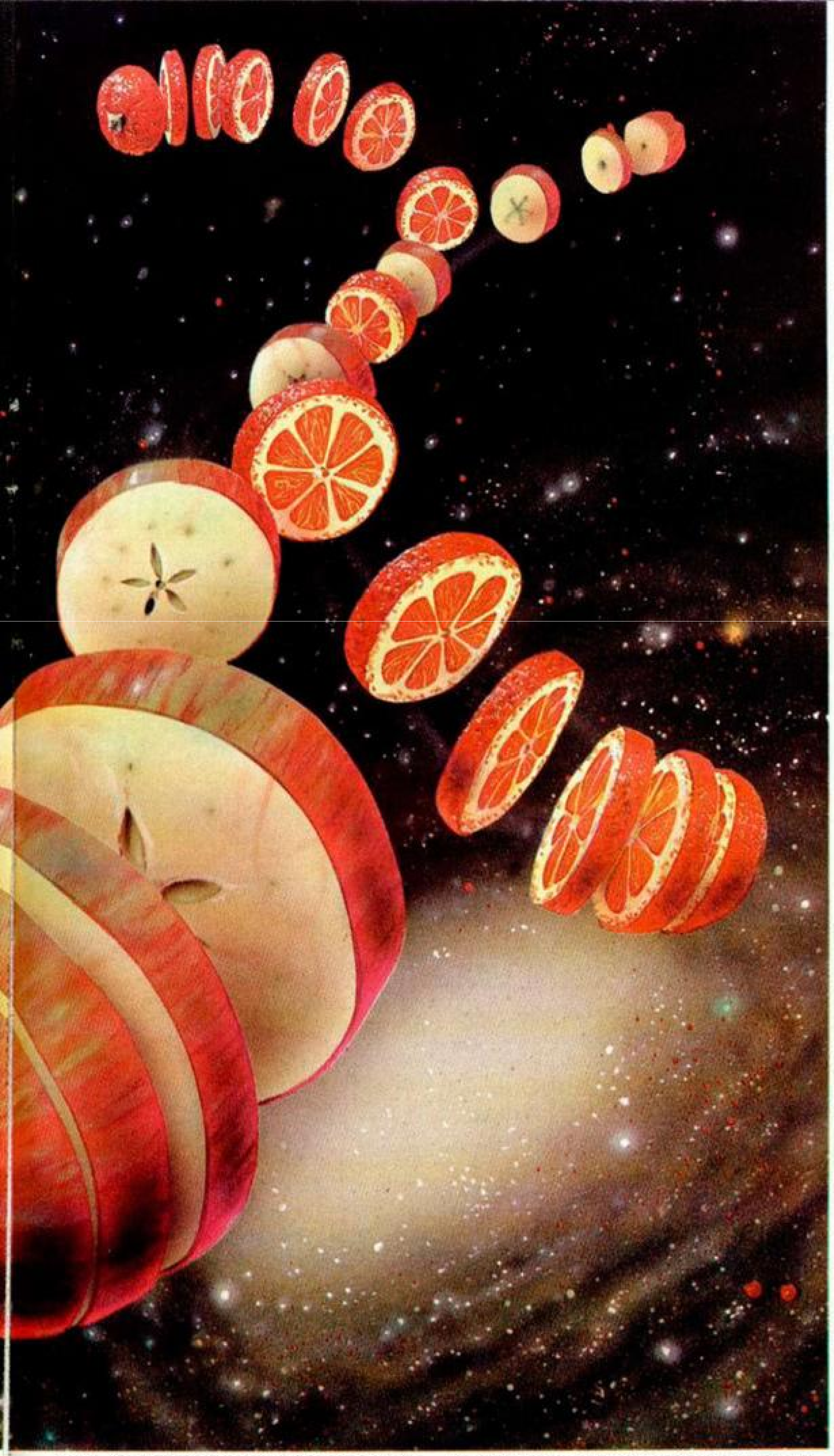


Fig. 1. Esquema del tratamiento de las interrupciones no enmascaradas NMI.



mas. Se vio la necesidad de que el software no las bloqueara, en primer lugar por su prioridad, y en segundo lugar porque un bloqueo involuntario dejaría sin atender las peticiones consideradas urgentes. Cuando un periférico interrumpe por NMI, inmediatamente el microprocesador salta a la dirección fija ØØ66 H (102 Decimal). A partir de la citada dirección, ha de estar contenida la subrutina que trate este tipo de interrupciones. Dicha subrutina ha de concluir con la instrucción de retorno de interrupción no enmascarable RETN.

La entrada NMI activa su interrupción correspondiente cuando este terminal es llevado a nivel bajo, disparándose justo con el flanco de bajada de la señal.

Mientras se está atendiendo una NMI, pueden llegar más flancos de bajada por el terminal citado, pues bien, cada flanco dará lugar a un nuevo salto a la dirección ØØ66H, aunque no se haya concluído el proceso de una interrupción anterior. Esto da lugar a que el stack se llene inadecuadamente con direcciones de retorno de la propia subrutina de interrupción, siendo únicamente válida la acción del último flanco. Para evitar esto, hay que preveer que el periférico que interrumpe, entregue una señal limpia con un único flaco de bajada. Por esta razón, es desaconsejable el uso directo sobre la entrada NMI de pulsadores o interruptores, pues sus microrrebotes producirían un número indefinido de flancos, generando los efectos negativos citados. Esto puede quedar resuelto, por ejemplo, con la conexión intermedia de un monoestable.

En el Spectrum, la NMI se encuentra bloqueada mediante un artificio software, el cual bifurca generalmente hacia la dirección ØØØØH, produciéndose la destrucción inmediata del contenido de la memoria, ya que aquí es donde se encuentra la rutina de inicialización del Spectrum. Próximamente propondremos varios métodos para poder rehabilitar esta interrupción tan importante.

La entrada de NMI se prolonga desde el Z-80 hacia el conector trasero del Spectrum por el terminal 14A, por donde se genera la interrupción llevándolo a masa.

La entrada INT activa las interrupciones enmascarables, las cuales en el Z-80 alcanzan una notable potencia.

INT (Interrupción enmascarable)

La interrupción INT puede presentar hasta tres modos diferentes, denominados IMØ (INTERRUPT MODE), IM-1 e IM-2. Cada uno de estos modos han de ser previamente predeterminados por software mediante las instrucciones assembler IMØ, IM1 e IM2 que se corresponden con los modos citados.

Las interrupciones enmascarables sí pueden ser deshabilitadas por software siempre que el programador lo considere necesario, y cuantas veces sea preciso. Para ello existen dos instrucciones assembler: EI (ENABLE INTERRUPT), que posibilita las peticiones de interrupción, DI (DISABLE INTERRUPT), que bloquea la entrada de este tipo de interrupciones. Así pues, ca-

da vez que se desee el uso de interrupciones enmascarables, habrá que poner la instrucción EI, la cual las posibilitará hasta que se ejecute su instrucción contraria DI. Estas instrucciones no tienen acción sobre la interrupción no enmascarable NMI.

El periférico que interrumpe por INT, ha de introducir un nivel bajo por el conector trasero, concretamente por el terminal 13A. Similarmente a lo dicho con la NMI, también se puede generar una interrupción INT mediante un pulsador o un interruptor que unan el citado terminal con masa para meter un cero lógico. Cuando el pulsador esté abierto, el nivel lógico de INT será un uno proporcionado por una resistencia conectada a +5V que se halla en la tarjeta del Spectrum.

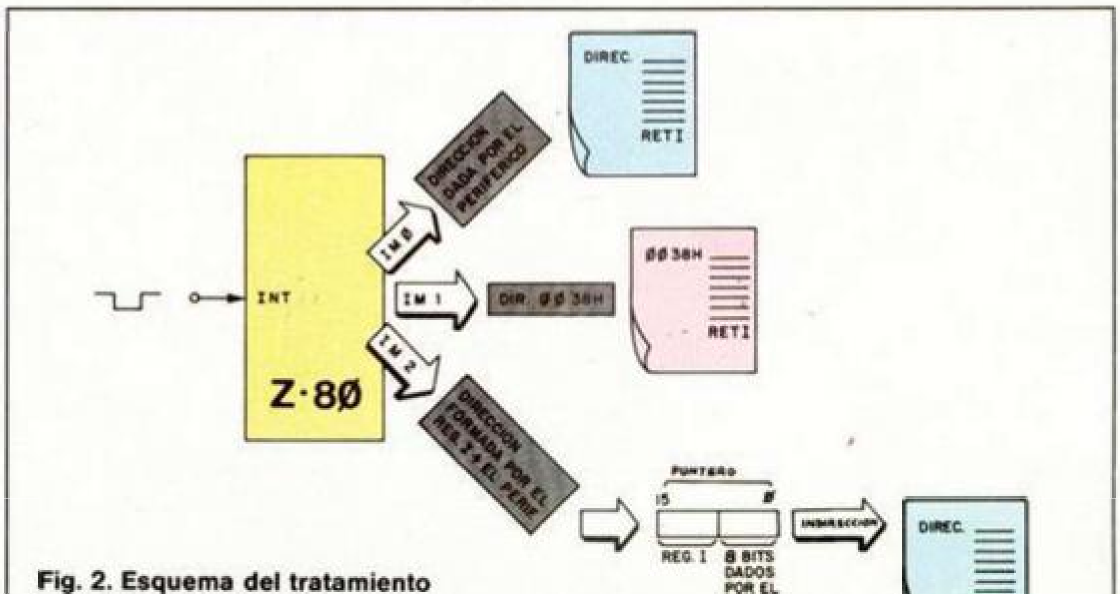
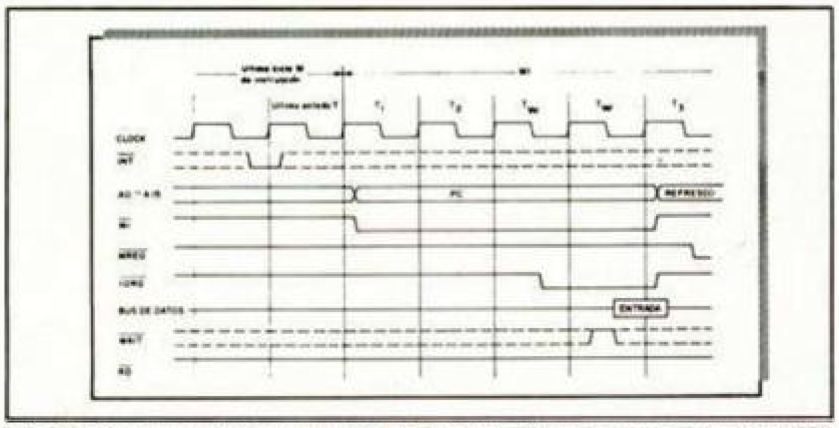
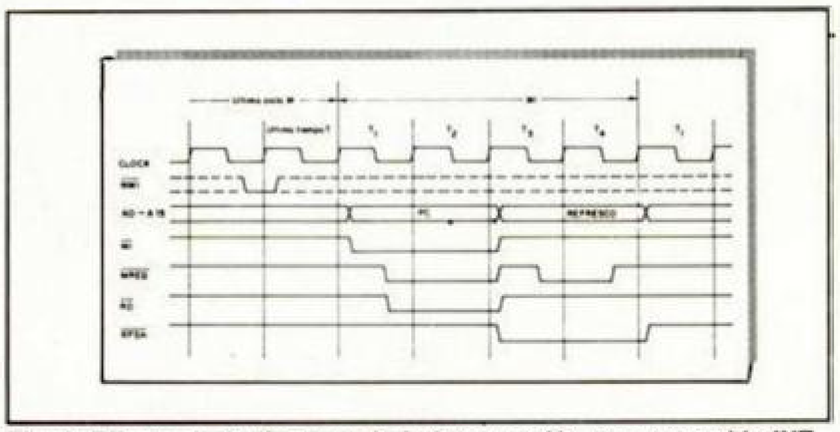


Fig. 2. Esquema del tratamiento de las interrupciones enmascarables en sus tres modalidades.


Fig. 3. Diagrama de tiempos de la interrupción no enmascarable NMI. La interrupción se produce cuando NMI pasa a nivel bajo.


Fig. 4. Diagrama de tiempos de la interrupción enmascarable INT. Tw indica que se piensen dos ciclos de reloj en el proceso.

Modos de interrupción enmascarable

Veamos a continuacioin los tres modos de interrupción INT mencionados.

MODO Ø

Este modo está copiado del precursor del Z-80, el 8Ø8Ø. En este modo de interrupción el periférico coloca al interrumpir en el bus de datos un byte que habrá de ser alguna de las instrucciones RESTART. Este tipo de instrucciones son de un único byte, y conllevan un salto absoluto predeterminado a las primeras direcciones de la página cero, en donde se hallará una instrucción de salto absoluto de tres bytes, que es donde estará ubicada la subrutina de tratamiento de interrupciones. En el Spectrum este modo es inusable, a no ser que se sustituya el contenido de la ROM, porque todas las direcciones de las instrucciones restart están ocupadas. Tras un reset, el Z-80 asume automáticamente el modo cero.

MODO 1

Este modo es similar al de NMI, únicamente cambia la dirección de salto, que pasa a ser ØØ38H (56 decimal). Este es el modo habitual del Spectrum, que usa las interrupciones para el tratamiento del teclado cuya subrutina está colocada a partir de la citada dirección.

MODO 2

Este es el más potente de los tres, está concebido para dar las máximas prestaciones del sistema de interrupciones a los periféricos.

En este modo intervienen el registro I interno del Z-80 y el propio periférico. Tras la interrupción, el Z-80 conforma una dirección absoluta entre el contenido del registro I, el cual ha de ser previamente cargado con el valor deseado, y el byte que habrá de entregar el periférico por el bus de datos. El registro I proporciona la parte alta, y el periférico la parte baja. La dirección así formada es en realidad un puntero, el cual apuntará a un byte de memoria que conjuntamente con el siguiente conformarán la dirección absoluta en que se encuentra la subrutina de tratamiento de la interrupción.

Este método de obtener una dirección absoluta, denominado «indirección», es ampliamente utilizado en todos los microprocesadores. Un sencillo ejemplo ayudará a comprender este modo que, repetimos, es el más versátil: Supongamos que en el registro I existe el dato hexadecimal B7H, y el periférico que interrumpe entrega el byte ACH, pues bien, se nos forma B7ACH como dirección absoluta. El microprocesador tomará el contenido de la dirección B7ACH que suponemos es 5AH, también tomará el contenido de la dirección siguiente B7ADH que suponemos es un 6BH, la dirección efectiva a la cual saltará será, por tanto, la 6B5AH. Siempre el primer byte será el de la parte inferior de la dirección efectiva. En el Spectrum este tipo de interrupciones sí es posible, sólo que el byte de menor peso de la dirección del puntero será FFH, que es lo que entrega el ordenador al no haber ningún periférico que proporcione otro dato.

La instrucción de retorno de las interrupciones INT es RETI.

Profesor particular



REPRESENTACION GRAFICA

Arturo LOBO y J. J. LEON

El articulo que os ofrecemos esta semana se sale un poco de nuestra linea habitual de ofreceros programas didácticos y educativos, programas que enseñan cosas.

Este os será de mucha utilidad cuando encontréis funciones extrañas o trascendentes cuya representación gráfica o resolución sean difíciles de obtener.

Dibujar en la pantalla la función que le pidáis, que no tiene por que ser explicita, y=f(x), sino que puede venir dada también en forma implícita f(x,y)=0, por ejemplo, sen y+y+x²=0 que no se puede dibujar sin la ayuda de un ordenador, por que es trascendente.

Le tenéis que dar entre qué valores de x queréis que os lo represente y un valor inicial de y que utiliza para resolver la función por métodos iterativos.

También os permite representar una matriz de puntos discretos que os interesará ver en la pantalla. Por ejemplo, podrían ser distintos puntos espacio-tiempo que podrían haber obtenido en un experimento de física sobre el movimiento de un cuerpo.

Esta matriz puede tener cualquier nombre, salvo v, q ó z.

Es una matriz de dos columnas, la primera será la de la variable del eje x que tiene que estar ordenada, la x de un punto mayor que de la del anterior.

La representación que aparece en la pantalla es realmente espectacular, el programa elige por sí solo, y dibuja la escala de las x y de las y, aunque os permite seleccionar a vosotros la escala. Dibuja cada unidad con el numerito que le corresponde debajo.

No utiliza los números del Spectrum, sino otros más pequeños que dan mayor vistosidad.

Este programa no puede correr en un Spectrum de 16k porque utilizan una rutina máquina que se ubica en la parte alta de la memoria por encima de la dirección 65000.

También conviene advertir que hay que tener cuidado al introducir la función, no olvidar ningún paréntesis, por ejemplo, pues el programa lo admitirá aunque dará error al ejecutarlo.

También hay que tener en cuenta que las funciones implícitas pueden tener ninguna o varias soluciones para un valor de x. El programa sólo halla una de ellas, o sea, sólo dibuja una de las formas de la función si tiene varias. Podría dibujar la otra escogiendo un valor inicial apropiado.

TODAS LAS LETRAS MAYUSCULAS SUBRAYADAS DEBERAN TECLEARSE EN MODO GRAFICO

```
6997 DEF FN x(x)=(x-origx)/unix*16+orx
6998 DEF FN y(y)=(y-origy)/uniy*16+ory
6999 DEF FN E(X)=INT (LN ABS X/LN 10)
7000 GO TO 7500
7002 LET dimq=0: LET dimw=0: LET dime=0
7004 RESTORE 7020
7005 FOR i=1 TO 8: READ m$: BEEP i/(i+15),2*i: PRINT AT 2*(i+(I<>1)),7;m$: NEXT i
7010 LET menu1=CODE INKEY$-48: IF (menu1>0)*(menu1<8)=1 THEN GO TO 7025
7011 GO TO 7010
7012 GO TO 7011+menu1+(menu1=1)
7013 LET orden=(menu1=2): GO SUB 7400: RETURN
7014 LET orden=2: GO SUB 7400: RETURN
7015 RETURN
7016 LET todo=1: GO SUB 7380: RETURN
7017 LET todo=0: GO SUB 7380: GO TO 7019
7018 LET todo=1: GO SUB 7380: GO TO 7019
7019 LET todo=0: GO SUB 7380: RETURN
7020 DATA "MENU DE ENTRAMADOS","1-PLOTEADO SIMPLE","2-PLOTEADO DOBLE","3-PLOTEADO TRIPLE","4-SIN ENTRAMADO","5-DOBLE RAYADO","6-RAYADO SIMPLE","7-PUNTO RAYA"
7021 DATA "INPUT DE FUNCIONES","1-  y=f(x)","2-  f(x,y)=0","3-MATRIZ DE PUNTOS","4-ELCCION DE ESCALA","5-   SALIR"
7026 LET PASA=-1
7027 DIM q(50,2): DIM z(50,2): DIM w(50,2): DIM e(50,2)
7030 CLS : RESTORE 7021: LET PASA=PASA+1: FOR i=1 TO 6: READ m$: BEEP i*3/4/(i+10),20-2*i: PRINT AT 2*(i+(I<>1))+(I<>1),8;m$: NEXT i
7035 LET menu2=CODE INKEY$-48: IF (menu2>0)*(menu2<6)=1 THEN CLS : GO TO 7035+5*menu2-(MENU2=5)
7036 GO TO 7035
7037 DEF FN f(X,y)=VAL q$
7040 INPUT "mete f(x)=", LINE q$: LET q$=q$+"-y": GO TO 7060
7045 INPUT "METE f(X,y)="; LINE q$: GO TO 7060
7050 INPUT "COMO SE LLAMA LA MATRIZ ? ";s$
7052 INPUT "CUANTOS PUNTOS HAY ? ";DIM1: DIM v(dim1,2): FOR I=1 TO DIM1: FOR j=1 TO 2: LET V(I,J)=VAL (s$+"(i,j)"): NEXT j: NEXT i
7053 GO TO 7060
7055 CLS : INPUT "ORIGEN EN X =";origx: INPUT "UNIDAD PARA X =";unix: INPUT "ORIGEN EN Y =";ORIGY: INPUT "UNIDAD PARA Y =";UNIY
7059 GO TO 7100
7060 IF menu2>2 THEN GO TO 7075
7061 INPUT "LIMITE INFERIOR DE X = ";x0: INPUT "LIMITE SUPERIOR DE X = ";x1: LET dim1=21: DIM v(21,2): INPUT "VALOR INICIAL PARA Y = ";Y0
7063 LET y=y0: FOR i=1 TO 21: LET x=x0+(i-1)/20*(x1-x0): LET y0=y: GO SUB 7270: IF cont<50 THEN LET v(i,1)=x: LET v(i,2)=y: NEXT i: GO TO 7075
7065 GO TO 7030
7075 FOR I=1 TO DIM1: FOR j=1 TO 2: GO SUB 7080+pasa: NEXT j: NEXT i: GO TO 7100
7080 LET q(i,j)=v(i,j): LET dimq=dimq+.5: RETURN
7081 LET w(i,j)=v(i,j): LET dimw=dimw+.5: RETURN
7082 LET e(i,j)=v(i,j): LET dime=dime+.5: RETURN
7100 LET orx=52: LET ory=11: LET bor=0: GO SUB 7430: GO SUB 7012
7105 LET dim=dimq: FOR i=1 TO dim: FOR j=1 TO 2: LET z(i,j)=q(i,j): NEXT j: NEXT i: IF menu2<>4 THEN GO SUB 7300
7110 GO SUB 7200: LET direc=50000: GO SUB 7220: GO SUB 7230: PRINT #1;"ENTER PARA CAMBIAR DE ESCALA"
7120 PAUSE 0: LET menu2=4: GO TO (7055-25*(CODE INKEY$<>13))
7200 LET ejex=1: FOR j=0 TO 10 STEP 2: LET num=origx+j*unix: LET ploy=ory-7: LET plox=orx+16*j: GO SUB 7250: NEXT j
7205 LET ejex=0: FOR j=0 TO 10: LET num=origy+j*uniy: LET plox=orx-2: LET ploy=ory+16*j-3*(j>0): GO SUB 7250: NEXT j
7210 RETURN
7220 FOR i=1 TO dim: LET xrt=FN x(z(i,1)): POKE direc+2*i,xrt*(xrt<=255)*(xrt>=0)+255*(xrt>255): NEXT i
7225 FOR i=1 TO dim: LET yrt=FN y(z(i,2)): POKE direc+2*i+1,yrt*(yrt<=175)*(yrt>=0)+175*(yrt>175): NEXT i
7226 RETURN
7230 PLOT OVER bor;PEEK (direc+2),PEEK (direc+3): FOR i=2 TO dim: DRAW OVER bor;PEEK (direc+2*i)-PEEK (direc+2*i-2),PEEK (direc+2*i+1)-PEEK (direc+2*i-1): NEXT i: RETURN
7250 LET num1=num: GO TO 7255
7251 LET num1=INT (num/10↑FN e(unix*(ejex=1)+uniy*(ejex=0))+.001)
7255 LET a$=STR$ num1: LET xp=plox-(5*(ejex=0)+2.5*(ejex=1))*LEN a$: FOR i=1 TO LEN a$: LET xp=FN v(xp,ploy,(CODE (a$(i))-48)*(a$(i)<>"-")*(a$(i)<>".")+10*(a$(i)=".")+11*(a$(i)="-")): NEXT i
7260 RETURN
7270 LET cont=0: LET y1=y0
7272 LET CONT=CONT+1: LET y2=y1-FN f(x,y1)/(FN f(x,y1+1e-3)-FN f(x,y1)+1e-6)/1000
7273 IF cont>50 THEN PRINT "PARA x=";x;" NO HAY SOLUCION": PAUSE 300: RETURN
7275 IF ABS (y2-y1)>ABS y2/200+1e-3 THEN LET y1=y2: GO TO 7272
7280 LET y=y2: RETURN
7300 LET xmax=z(1,1): LET xmin=z(1,1): LET ymin=z(1,2): LET ymax=ymin: FOR i=1 TO dim
7305 LET xmax=xmax+(z(i,1)-xmax)*(z(i,1)>xmax)
7310 LET xmin=xmin+(z(i,1)-xmin)*(z(i,1)<xmin)
7315 LET ymin=ymin+(z(i,2)-ymin)*(z(i,2)<ymin)
7320 LET ymax=ymax+(z(i,2)-ymax)*(z(i,2)>ymax)
7325 NEXT i: LET dx=xmax-xmin: LET dy=ymax-ymin
7330 LET d0x=ABS xmax: LET d0x=d0x+(ABS xmin-d0x)*(d0x<ABS xmin)
7335 LET d0y=ABS ymax: LET d0y=d0y+(ABS ymin-d0y)*(d0y<ABS ymin)
7336 LET d0x=d0x*(d0x<2*dx): LET d0y=d0y*(d0y<2*dy)
7337 LET d1x=dx: LET d1x=d1x+(d0x-d1x)*(d1x<d0x)
7339 LET d1y=dy: LET d1y=d1y+(d0y-d1y)*(d1y<d0y)
7341 LET d1x=1.15*d1x: LET unix=(INT (d1x/10↑FN e(d1x))+1)*10↑(FN e(d1x)-1)
7343 LET d1y=1.15*d1y: LET uniy=(INT (d1y/10↑FN e(d1y))+1)*10↑(FN e(d1y)-1)
7345 LET origx=(INT (xmin/unix)*unix)*(d0x<>0)+(d0x=0)*(INT (xmin/10↑FN e(unix))*10↑FN e(unix))
7350 LET origy=(INT (ymin/uniy)*uniy)*(d0y<>0)+(d0y=0)*(INT (ymin/10↑FN e(uniy))*10↑FN e(uniy))
7355 RETURN
7380 LET hor=0: FOR i=1 TO 10: LET ply=ory+5
7382 FOR k=1 TO 10: FOR l=0 TO 1: LET lon=5-2*(l=1): LET plx=orx+i*16: LET bor=(todo=0)*(l=1): GO SUB 7450: LET ply=ply+lon+4: NEXT l: NEXT k
7385 NEXT i
7387 LET hor=1: FOR i=1 TO 10: LET plx=orx+5
7389 FOR k=1 TO 10: FOR l=0 TO 1: LET lon=5-2*(l=1): LET ply=ory+i*16: LET bor=(todo=0)*(l=1): GO SUB 7450: LET plx=plx+lon+4: NEXT l: NEXT k
7391 NEXT i: RETURN
7400 IF orden>=0 THEN LET orix=orx: LET oriy=ory: GO SUB 7420
7403 IF orden>=1 THEN LET orix=orx-8: LET oriy=ory: GO SUB 7420: LET orix=orx: LET oriy=ory-8: GO SUB 7420
7408 IF orden>=2 THEN FOR k=0 TO 1: FOR l=0 TO 1: LET orix=orx+(-12+8*l)*k: LET oriy=ory+(-12+8*l)*(k=0): GO SUB 7420: NEXT l: NEXT k: RETURN
7410 RETURN
7420 FOR i=1 TO 10: FOR j=1 TO 10: PLOT OVER bor;orix+16*i,oriy+16*j: NEXT j: NEXT i: RETURN
7430 LET plx=orx: LET ply=ory: LET lon=8*20: LET hor=1: GO SUB 7450: LET hor=0: GO SUB 7450
7435 LET ply=ory-1: LET lon=2: FOR i=1 TO 10: LET plx=orx+i*16: GO SUB 7450: NEXT i: LET hor=1: LET plx=orx-1: FOR i=1 TO 10: LET ply=ory+i*16: GO SUB 7450: NEXT i: RETURN
7450 PLOT OVER bor;plx,ply: DRAW OVER bor;lon*(hor=1),lon*(hor=0): RETURN
7500 CLEAR 64999: RESTORE 7550
7510 LET SUM1=0: FOR i=65000 TO 65077: READ a: LET sum1=sum1+a: POKE i,a: NEXT i
7520 FOR i=65400 TO 65431: READ a: POKE i,a: LET sum1=sum1+a: NEXT i
7530 FOR i=65480 TO 65527 STEP 4: FOR j=0 TO 2: READ a: LET sum1=sum1+a: POKE i+j,a: NEXT j: POKE i+3,0: NEXT i
7550 DATA 205,148,30,245,205,153,30,197,205,153,30,197,225,209,123,198,5,95,241,229,38,255,203
7560 DATA 39,203,39,198,200,111,78,44,86,44,70,225,96,121,205,120,255,205,120,255,122,205,120
7570 DATA 255,205,120,255,124,205,120,255,125,198,5,6,0,79,205,43,45,1,0,0,205,43,45,1,1,0,197,205,43,45,193,201
7580 DATA 6,4,197,203,127,40,11,67,77,245,213,229,205,223,34,225,209,241,44,203,7,193,16,234,79,125,214,4,111,121,29,201
7590 DATA 105,153,96,38,34,32,105,36,240,105,41,96,38,111,32,248,225,224,104,233,96,241,36,64,105,105,96,105,113,96,0,0,32,0,112,0
7595 GO TO 7002
7598 DEF FN v(x,y,n)=x+y*n↑USR 65000
```

EL SONIDO EN EL 128 K (y II)

Rafael Prades

Continuando con el comando «PLAY» se explica, en este último capítulo, la forma de seleccionar los diversos tipos de envolventes, para ser utilizadas en efectos sonoros, o simplemente, cómo controlar la duración y volumen de cada nota.



En términos musicales, no basta sólo con dominar a la perfección el arte de combinar las distintas notas; casi tan importante como esto es controlar la duración de cada una de ellas.

Por defecto, la duración relativa de cada nota es equivalente a la de una «negra», pero podemos programar desde una «semicorchea» (mínima duración) hasta una «redonda» (máxima duración), pasando por las «corcheas», «negras» y «blancas». La relación entre ellas es doble o mitad, es decir, la *blanca* es la mitad que la *redonda*, la *negra* es la mitad que la *blanca* y así sucesivamente.

Cada una de ellas está identificada por un código que se relaciona en la Tabla I. Cuando el sistema operativo lo decodifica, todas las notas siguientes tienen la misma duración hasta que se recibe un nuevo código.

Ejemplo:

```
LET a$="3cde7fgab"
PLAY a$
```

donde las notas «do», «re», y «mi» tendrán la duración de una corchea y el resto (fa, sol, la y si) de una blanca.

Los códigos 1Ø a 12 corresponden a lo que en el lenguaje musical se denominan «tresillos», es decir, tres notas que se ejecutan en el tiempo de dos; por tanto, este código sólo afecta a las tres notas que le sigue.

Ejemplo:

```
LET z$="10efd"
PLAY z$
```

las notas «mi», «fa», y «re» se ejecutarán en el mismo tiempo que dos semicorcheas.

Los *silencios*, es decir, pausas entre la ejecución de notas se indican con el símbolo «&» y tienen una duración equivalente al código especificado.

Ejemplo:

```
LET u$="9df&7e&c"
PLAY u$
```

primero se ejecutan las notas «re» y

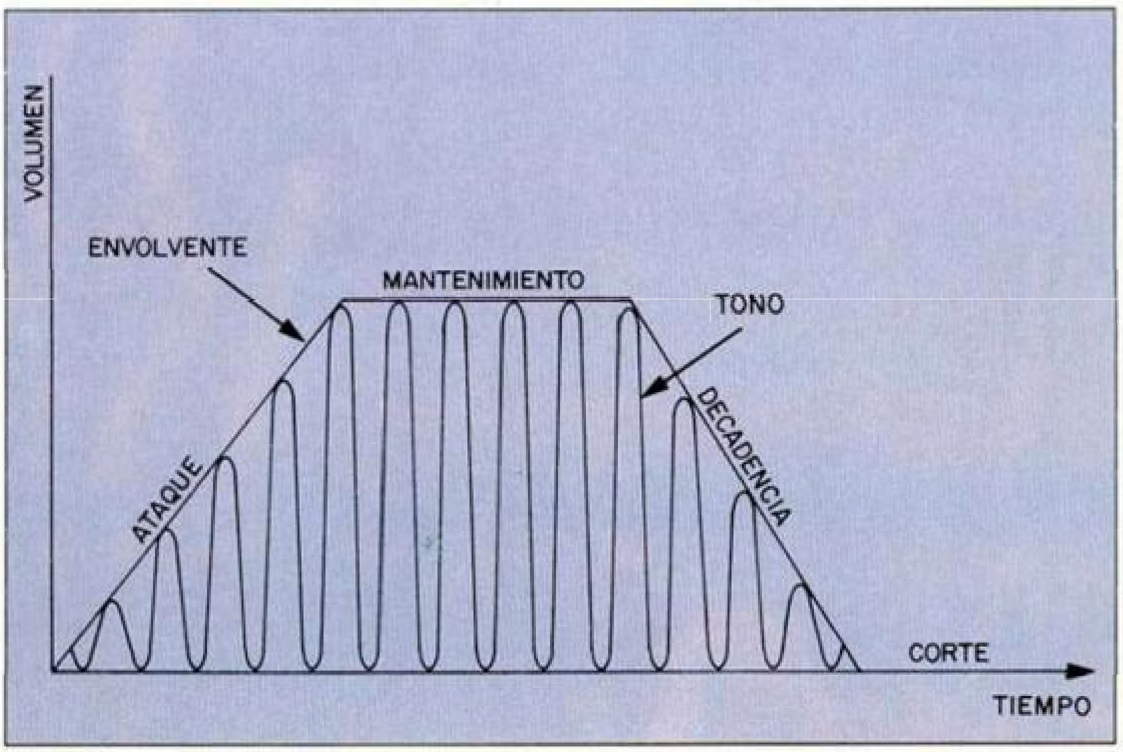


Fig. 1. Tono y envolvente.



«fa» seguidas de una pausa, con una duración cada una de ellas de una redonda y posteriormente las notas «mi», y «do» con una pausa intermedia correspondiente a una blanca.

El término *ligadura* que indica la ejecución de varias notas sin variaciones bruscas entre ellas, se especifica con el símbolo del subrayado.

Ejemplo:

```
LET f$="1_3dc"
PLAY f$
```

```
LET p$="043cdefgab"
```

ya que el sistema operativo interpretaría que deseamos seleccionar la octava «43» y se crearía una confusión al no existir ésta; por tanto, se solucionará el problema separando ambos conceptos (escala y duración) con el comando «N».

Ejemplo:

```
LET p$="04N3cdefgab"
PLAY p$
```

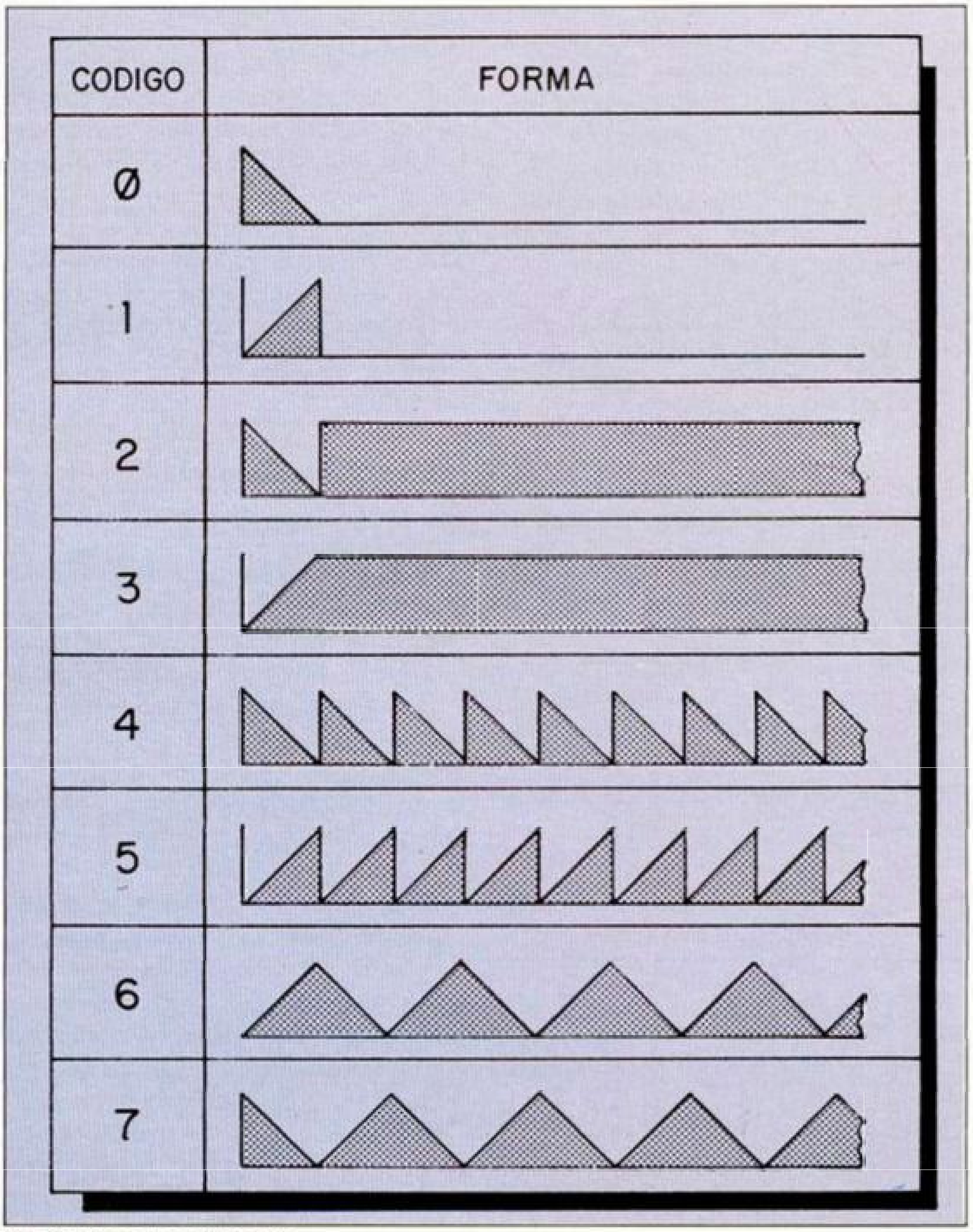


Fig. 2. Tipos de envolvente.

El comando «N»

Este no es propiamente un comando, ya que se utiliza como separador. Para explicar este concepto es mejor recurrir a un ejemplo.

Si deseamos seleccionar la octava «4» y posteriormente ejecutar la escala inferior completa, con la duración de una corchea, no podríamos utilizar la siguiente asignación:

Tempo

La velocidad de ejecución de una determinada pieza musical viene determinada por la cantidad de «negras» que se interpretan en un minuto (bpm). Si no se especifica ninguna cantidad el sistema operativo toma por defecto el valor 12Ø bpm.

Con el comando «T» se puede programar solamente la velocidad de eje-

cución del canal «A», pudiéndola variar entre 6Ø y 24Ø bpm.

Ejemplo:

```
LET g$="T8ØN7cdefagb"
PLAY g$
LET h$="T2ØØN7cdefagb"
PLAY h$
```

que interpreta la misma composición a dos velocidades diferentes.

Volumen

El volumen de las notas puede ser programado individualmente o en conjunto con el comando «V». La gama de intensidad sonora puede variar entre los valores «Ø» y «15», siendo este último el valor tomado por defecto.

El valor «Ø» corresponde al volumen más bajo (silencio) y por el contrario el «15» el más alto.

Ejemplo:

```
LET j$="Nlcdefgab7&"
FOR n=Ø TO 15
LET k$="V"+STR$ n+j$
PRINT "Volumen: ";n
PLAY K$
NEXT n
```

donde la escala musical es interpretada a diferente volumen.

Repetición

Un grupo de notas puede repetirse una sola vez o indefinidamente utilizando los símbolos de los paréstesis. Estos pueden hacerse de tres formas diferentes:

a) Repetición de un grupo determinado de notas.

Ejemplo:

```
LET w$="agbg(fde)"
PLAY w$
```

las últimas notas (fa, re y mi) se repiten.

b) Repetición de toda la cadena. Se omite el primer paréntesis.

Ejemplo:

```
LET m$="agbgfde)"
PLAY m$
```

c) Repetición indefinida de un grupo de notas.

Ejemplo:

```
LET o$="agbgfde))"
PLAY o$
```

Esta última opción es muy utilizada cuando se desea que un canal interprete, por ejemplo, las notas repetitivas de una composición.

Ejemplo:

```
LET d$="ddefdgfeg"
LET e$="01N1cde))"
PLAY d$,e$
```

la melodía repetitiva del canal «B» sólo terminará, si en la cadena que controla el canal «A», se añade el comando «H» al final

```
LET d$="ddefdgfegH"
```

Comentarios

Si el contenido de una cadena es muy largo y contiene muchos comandos, es posible que al cabo de un cierto tiempo no recordemos su significado; por tanto, utilizando el símbolo de admiración podemos incluir comentarios dentro de la cadena, que son ignorados por el sistema operativo y que nos permiten recordar cualqueir detalle de interés.

Ejemplo:

```
LET i$=iMelodia!cdegfecd
        icoro!(abgf)"
```

Envolvente

El «PSG» dispone de un generador de envolvente que varía el volumen de cada nota según se va ejecutando, de una forma similar a la modulación de amplitud (AM) utilizada en la transmisión radiofónica.

En la figura «1» se pueden observar las distintas partes de una envolvente: ataque, decadencia, mantenimiento y corta.

El *ataque* es el aumento gradual y suave de una nota, por el contrario la *decandencia* o *amortiguamiento* hace referencia al descenso gradual del volumen. En la zona de *mantenimiento* el volumen es constante y en la de *corte* termina la ejecución de la nota.

Con el comando «W» se pueden programar ocho tipos diferentes de envolvente (Ø a 7), las cuales se encuentran representadas en la figura «2» y Tabla II.

El comando «V» (volumen) tiene mayor prioridad que la selección de envolvente, por este motivo esta última sólo es efectiva en cadenas en las que no se programa el volumen.

El comando «U» selecciona el canal afectado por el generador de envolventes y el «X» determina la longitud de la envolvente; es necesario especificar un valor entre «Ø» y «65535», aunque por defecto se encuentra seleccionado este último valor.

Ejemplo:

```
1Ø LET r$="UX"
2Ø LET s$="ccefdeaab9&"
3Ø FOR n=Ø TO 7
4Ø IF n < 4 THEN LET t$=
   "1ØØØ": GO TO 6Ø
5Ø LET t$="3ØØ"
6Ø LET u$=r$+t$+"W"+STR$
   n+s$
7Ø PLAY u$
8Ø NEXT n
```

En la Tabla III se encuentran los valores que el 128 K selecciona por defecto.

TABLA I

NOMBRE	CODIGO
SEMICORCHEA	1
Idem con puntillo	2
CORCHEA	3
Idem con puntillo	4
NEGRA	5
Idem con puntillo	6
BLANCA	7
Idem con puntillo	8
REDONDA	9
TRESILLO DE SEMICORCHEAS	1Ø
Idem de CORCHEAS	11
Idem de NEGRAS	12

TABLA II

ENVOLVENTE	CODIGO
DECADENCIA Y CORTE	Ø
ATAQUE Y CORTE	1
DECADENCIA Y MANTENIMIENTO	2
ATAQUE Y MANTENIMIENTO	3
DECADENCIA REPETIDA	4
ATAQUE REPETIDO	5
REPETICION DE ATAQUE Y DECADENCIA	6
DECADENCIA Y ATAQUE REPETIDO	7

TABLA III

VALORES POR DEFECTO	
TIPO DE CANAL	TONO
OCTAVA	5
DURACION	NEGRA
VOLUMEN	15
TIEMPO (bpm)	12Ø
DURACION ENVOLVENTE	65535

MICRO Manía

Sólo para adictos

Os ofrecemos en esta sección, cedida por Micromanía, todo tipo de pokes y trucos que os ayudarán a conseguir mejores puntuaciones en los juegos, recordándoos que estamos abiertos a cualquier tipo de colaboración por vuestra parte. Para ello no tenéis más que enviar vuestros descubrimientos y aportaciones a MICROHOBBY, poniendo en el sobre «SECCION MICROMANIA».

MATCH POINT

Estaréis de acuerdo en que Match Point es uno de los mejores juegos de entre todos los que se han realizado para el Spectrum (para nosotros estaría entre los tres mejores).

Pues por si acaso lo tenéis olvidado en el fondo de un cajón, os ofrecemos un pequeño cargador con el que podréis realizar una interesante variación y que hará que volváis con el bello deporte de la raqueta.

Lo que tenéis que hacer es lo siguiente:

Teclead MERGE "" y ENTER.

Editad la línea 10 y cambiar el número 23313 por el 23320.

Quitad los tres últimos números de la línea 200 y teclead lo siguiente:

210 DATA 175,50,142,209,50,



153,209,195,0,99 y después ENTER. Luego corred y cargad el resto del programa.

Observaréis que la velocidad de los jugadores es prácticamente el doble de la normal, lo que hará que en cuartos de final y semifinales resulte un poco desesperante, pero en las finales la emoción crecerá enormemente.

SKOOL DAZE

En el número 54 de nuestra revista publicábamos algunas de las fechas solicitadas en este simpático juego con el fin de facilitaros ligeramente las cosas.

Pero no contento con esto, Aitor Loizaga nos ha escrito desde Vizcaya una amable carta en la que nos muestra casi la totalidad de dichas fechas, por lo que ahora aún os resultará más sencillo responder a las



preguntas de vuestros profesores.

Waterloo-1815
Borodino-1812
Eversham-1265
Bosworth-1485
Lexington-1775
Poitiers-1356
Hastings-1066
Grecy-1346
Gettysburg-1863
Lepanto-1571
Bannockburn-1314
Floden-1513
Culloden-1746
Trafalgar-1805
Yorktown-1781
San Jacinto-1830
Agincourt-1415

COMO MANEJAR LA PANTALLA DESDE CODIGO MAQUINA (y II)

Paco PORTALO

Seguramente muchos de vosotros estáis impacientes por saber cómo se anima una figura en pantalla, moviéndola pixel a pixel. Para conseguir esto, que no es precisamente labor de un solo día, primero es imprescindible coordinar las diversas posiciones de la memoria de pantalla.

La semana pasada iniciábamos la explicación de una rutina de coordenación que permitía relacionar las diversas columnas y filas de la pantalla en baja resolución (caracteres) con las correspondientes posiciones de memoria.

Ahora vamos a ver cómo se consigue el mismo objetivo pero a nivel de pixel, es decir, en alta resolución.

Coordenación por pixels

Vamos a examinar de nuevo la forma de componer la dirección de mememoria a partir de las coordenadas. (Ver Fig. 1.)

Fijaos que si dividiéramos por 8 la línea y la columna, nos quedaría lo mismo que vimos anteriormente (en código máquina podemos hacer esta operación haciendo tres desplazamientos de izquierda a derecha, mediante la instrucción SRL r).

Estudiemos pues los arreglos que tenemos que introducir. (Ver «desensamble de la rutina».)

Este programa nos pondrá un punto (pixel) en cualquier coordenada de la pantalla dada en los registros B y C. Es similar a la rutina de caracteres, con la inserción de nuevas instrucciones que estudiamos detalladamente ahora.

DESARROLLO DE LA RUTINA

LD H, # 4Ø

LD A,B

AND 7: Para coger la línea, que corresponde a los tres bits menos significativos de la coordenada dada en B.

ADD A,H

LD H,A: La incorporamos en su sitio definitivo.

SRL B: Dividimos por 8, con lo que la coordenada de línea del registro B nos queda en el formato del programa anterior. Fijaos que cada SRL B divide por 2.

LD A,B

AND 24

ADD A,H

LD H,A

LD A,B

AND 7

RRCA

RRCA

RRCA

LD L,A: Salvamos en L la fila.

LD A,C: Metemos en el acumulador la columna.

AND 7: Tomamos los tres bits bajos que nos definirán el pixel a pintar dentro del byte de columna.

LD B,A: Lo salvamos en B.

SRL C: Dividimos por 8 la columna una vez conocido el bit que tenemos que pintar, con lo que tendremos en el registro C lo mismo que en el programa anterior.

LD A,C: Lo metemos en A para operar.

ADD A,L: Lo sumamos al registro L donde tenemos ya la fila.

LD L,A: Lo recuperamos, con lo que tenemos formada la dirección en el registro doble HL.

Una vez calculada la dirección del byte donde se aloja el pixel a pintar, y, además de conocer cuál es (registro B), daremos una pequeña explicación del procedimiento a seguir.

Como sabéis, los bits están emplazados en cada byte desde el más significativo (a la izquierda) hasta el menos significativo (a la derecha), quiere decir esto que, para pintar el pixel Ø,Ø tendríamos que meter un 1 en el bit más significativo de la dirección de comienzo de pantalla (como la coordenación comienza en la parte superior de la pantalla, habría que meter 128 en la dirección # 4ØØØ).

BIT	7	6	5	4	3	2	1	0
PIXEL	0	1	2	3	4	5	6	7

La disposición de cada BIT dentro de un BYTE es justamente inversa desde el punto de vista de los pixels.

TRANSFORMACION DE UN BIT EN PIXEL

Nosotros tenemos en el registro B el bit, pero ojo, en modo de pantalla, es decir, que para pintarlo en su sitio correspondiente tendremos que hacer un pequeño truco:

INC B: Aumentamos el valor del registro B en 1.

XOR A: Ponemos el acumulador a Ø.

SCF: Esta instrucción nos pone el indicador de carry a 1.

C1 RRA, DJNZ C1: Este pequeño bucle nos girará de izquierda a derecha el acumulador tomando el carry tantas veces como indique el registro B.

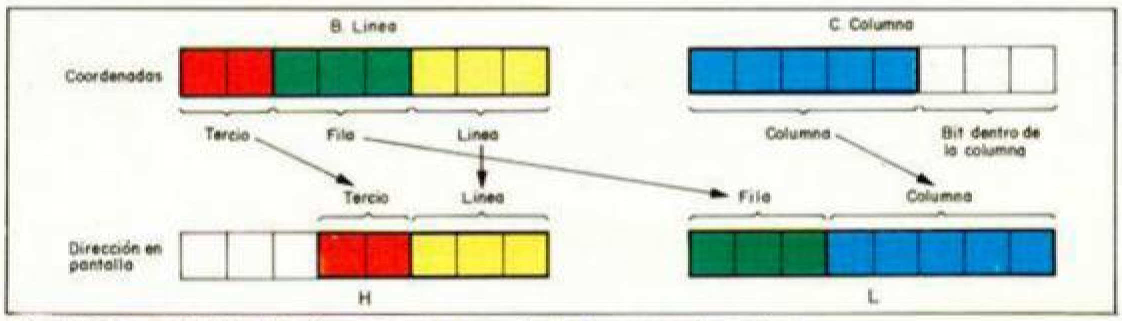


Fig. 1. Composición de direcciones a partir de las coordenadas.

En B tenemos un número entre 1 y 8, por lo tanto si hacemos un RRA, y, teniendo en cuenta que tenemos A a Ø y el carry a 1, se nos pondría el bit 7 de A a 1, y todos los demás a Ø. Si en B tuviéramos 8, haría 8 RRA, con lo que se nos pondría a 1 el bit Ø de A.

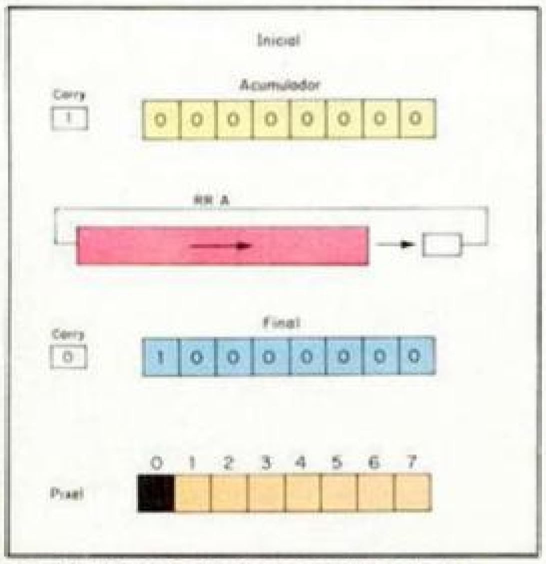


Fig. 2. Rotación de bytes utilizando la instrucción RRA.

SOBREIMPRESION DE VARIOS BITS EN UN SOLO BYTE

Como se sabe, los bits están agrupados de 8 en 8 en bytes. Cuando queremos alterar un determinado bit de un byte, hemos de hacerlo sin modificar el resto de los bits que lo componen.

OR (HL): Hacemos esta operación lógica del registro A con el contenido de la dirección HL con objeto de no borrar los pixels que ya estuvieran pintados.

LD (HL), A: Pintamos el pixel correspondiente.

RET: FIN.

Si os fijáis, esta rutina es totalmente reubicable, quiere esto decir que se puede poner perfectamente en cualquier parte de la memoria RAM y funcionará correctamente. Se debe a que no utilizamos variables fijas ni saltos absolutos; esta condición es muy importante si trabajamos en código máquina, pero no siempre es posible hacerlo. Para probarla podéis teclear el siguiente programa máquina en el que se llama a C0 como subrutina.

DEMOSTRACION

```
10        LD   B,191
20        LD   C,0
30 E0     PUSH BC
40        CALL C0
50        POP  BC
60        INC  C
70        JR   NZ,E0
80        DJNZ E0
```

(CO sería en este programa la dirección donde se encuentra la subrutina de coordenación).

Y comparar la rapidez con el siguiente programa Basic.

```
10 FOR B=0 TO 175: FOR C=0 TO
255: PLOT C,B: NEXT C: NEXT B
```

Para aquellos que quieran utilizar esta rutina desde Basic, a continuación está el listado del programa

```
 10 CLEAR 59999
 20 FOR I=0 TO 49
 30 READ A
 40 POKE 60000+I,A: NEXT I
 50 INPUT "LINEA: ";B,"COLUMNA: "
;C
 60 POKE 60001,C: POKE 60002,B
 70 RANDOMIZE USR 60000
 80 STOP
 90 DATA 1,0,0,38,64,120,230,7,
132,103,203,56,203,56,203,56,120
,230,24,132,103,120,230,7,15,15,
15,111,121,230,7,71,203,57,203,5
7,203,57,121,133,111,4,175,55,31
,16,253,182,119,201
```

Tenéis que tener en cuenta que en este programa las coordenadas Ø,Ø están en la parte superior izquierda de la pantalla, al contrario que el PLOT en el Basic. Si queréis que funcione como éste, sólo tendréis que hacer unos pequeños arreglos que os dejamos como ejercicio para que vayáis practicando. También sería interesante que le pusierais topes para que no fuera posible introducir una línea superior a 191.

DESENSAMBLE DE LA RUTINA DE COORDENACION POR PIXELS

```
10        LD   H,#40       190        LD   A,C
20        LD   A,B         200        AND  7
30        AND  7           210        LD   B,A
40        ADD  A,H         220        SRL  C
50        LD   H,A         230        SRL  C
60        SRL  B           240        SRL  C
70        SRL  B           250        LD   A,C
80        SRL  B           260        ADD  A,L
90        LD   A,B         270        LD   L,A
100       AND  24          280        INC  B
110       ADD  A,H         290        XOR  A
120       LD   H,A         300        SCF
130       LD   A,B         310 C1     RRA
140       AND  7           320        DJNZ C1
150       RRCA             330        OR   (HL)
160       RRCA             340        LD   (HL),A
170       RRCA             350        RET
180       LD   L,A
```

CONSULTORIO

Radio Spectrum

Otra vez me dirijo a ustedes para intentar aclararles sobre la pregunta del consultorio donde un lector quiere saber si es posible conectar varios Spectrum vía radio. No sólo es posible sino que, de hecho, ya se está llevando a cabo hace mucho tiempo.

El alcance de dichas transmisiones depende del equipo radio-transmisor. Por mi parte, poseo QSLs de más de 10 naciones de Europa, e incluso, de Canadá que confirman mis contactos en modo RTTY.

Las frecuencias usadas son las siguientes: Bandas de 10, 15, 20, 40 y 80 metros. Yo transmito en la banda de 20 m. exactamente en la frecuencia de 14.080 a 14.100 Mhz.

Antonio BLANCO - Valencia

☐ Le agradecemos muy sinceramente su aclaración y animamos a otros lectores a que nos escriban contándonos sus experiencias en este terreno. No obstante, la polémica que surgió versaba sobre la comunicación por Banda Ciudadana (11 metros) donde las condiciones de QRM en grandes ciudades hacen imposible conseguir resultados satisfactorios en la mayoría de los casos.

Para aquellos de nuestros lectores que no estén versados en temas de radio-afición, incluimos un pequeño glosario del «argot» empleado:

- QSL: Tarjeta postal que se utiliza entre radioaficionados para confirmar los contactos.
- RTTY: Modo de transmisión que consiste en modular la frecuencia portadora con datos de ordenador.
- CB: (Banda Ciudadana), Banda de frecuencias comprendidas entre 26.965 y 27.405 Mhz. (11 metros) utilizada profusamente por una inmensa mayoría de radioaficionados, generalmente, sin licencia y con equipos de baja calidad, lo que hace que las condiciones de transmisión sean pésimas.
- QRM: Abreviatura perteneciente al código «Q» (de gran uso en radioafición) que se refiere al ruido de fondo existente en una transmisión y producido por interferencias de otras emisoras.

Avería fantasma

A mi ZX-Spectrum 48K le pasa que, a veces, sin tocarle siquiera, se borra todo el programa o se bloquea. Esto ocurre cada dos por tres, tanto cuando estoy jugando como cuando estoy haciendo algún programa (no veáis que gracia me hace cuando ésto ocurre).

¿A qué se debe la avería? ¿Podría arreglarlo yo mismo?

Oscar PARRA - Vizcaya

☐ Lamentablemente, es imposible determinar por correro la avería que sufre un ordenador. No obstante, intentaremos darle algunas pautas a seguir.

Lo primero que hay que comprobar es la alimentación, es posible que una interrupción interna en el cable sea la causa de todos sus problemas. El segundo paso es comprobar si la clavija que conecta con el ordenador tiene algún muelle flojo o sucio. Si tras estas posibilidades no ha resuelto el problema, puede abrir la fuente de alimentación y el ordenador (esto ya se pone más delicado) para buscar algún cable mal soldado o alguna conexión mediosuelta. Si llegado este punto, no ha dado con la avería, es mejor que ponga su ordenador en manos de un

PROFESOR PARTICULAR

Por un error de impresión, parte del listado de «Polinomios III» publicado en la sección "Profesor Particular» del número 61 de Microhobby, salió borroso, por lo que hemos creido conveniente repetir las partes menos legibles, que os ofrecemos a continuación.

```
5240 LET col2=3+(i>=gra3): GO SU
B 4260
5510 LET flm=0: PRINT "-Estos so
n los resultados:";''''"  f(";d$;
")=";s$''"   f'(";d$;")=";t$''"
f''(";d$;")=";u$''
5514 IF coc THEN LET pol=1
5515 FOR i=1 TO 21: FOR j=1 TO c
oc+1: LET p(i,1+pol+j)=g(i,1,j):
 NEXT j: NEXT i
6000 GO SUB 4480: PRINT #0;AT 1,
0;"Ahora dibujamos la funcion:"
6010 GO SUB 6750: GO SUB 4480: G
O TO 200
6500 CLS : PRINT "ASINTOTAS";AT
3,2;"-Horizontales: ";: LET i=21
6510 IF NOT (g(i,1,1) OR g(i,1,2
)) THEN LET i=i-1: GO TO 6510
6520 IF g(i,1,1)*g(i,1,2) THEN L
ET aa=g(i,1,1)/g(i,1,2): PRINT "
y=";aa: GO TO 6540
6530 PRINT "NO HAY": LET aa=PI
6540 PRINT AT 6,2;"-Oblicuas: ";
6550 IF g(i,1,1)<>0 AND aa=PI AN
D i>1 THEN IF g(i-1,1,2)<>0 THEN
 GO TO 6570
6560 LET bb=PI: PRINT "NO HAY":
GO TO 6590
6570 LET bb=g(i,1,1)/g(i-1,1,2):
 LET cc=0: LET dd=g(i-1,1,1)/g(i
-1,1,2): IF i>2 THEN LET cc=dd-b
b*g(i-2,1,2)
6580 PRINT bb;d$;: IF cc THEN PR
INT CHR$ (43+2*(cc<0));: PRINT A
BS cc
6590 PRINT AT 9,2;"-Verticales:
";: IF z4=0 THEN PRINT "NO HAY":
 GO TO 6610
6600 FOR i=1 TO z4: PRINT d$;"="
;e(i+1+z1+z2+z3)'TAB 15;: NEXT i
```

especialista; no olvide llevarle, también, la fuente de alimentación. Tenga cuidado, si abre el ordenador, porque las cintas que conectan el teclado se rompen «sólo con mirarlas»; en ningún caso lo abra mientras esté conectado.

«Programas de lectores»

Hace algunas semanas vengo observando que no incluyen Uds. la sección «PROGRAMAS DE LECTORES» en la revista. Quisiera saber si ya no piensan publicar más esta sección o tan solo es una pequeña pausa para seleccionar los mejores programas, ya que tengo un programa para enviarles, esta vez de utilidades.

Pere BUTJOSA - Barcelona

□ Debido a la inmensa cantidad de programas que habían invadido nuestra revista (y nuestra redacción), hemos decidido seleccionarlos más, por lo que, cada semana, daremos un solo programa o, a lo sumo, dos; pero de una calidad muy superior a los que veníamos dando hasta ahora. Creemos que nuestros lectores nos lo agradecerán, ya que muchos se habían quejado de que pasaban horas tecleando un programa para que el final, en su opinión, no valiera la pena.

Quizás la idea no les parezca tan buena a quienes nos enviaban sus programas (en cierto sentido hemos «subido el listón»), pero contamos con que, a la larga, esta decisión sirva para estimular a los programadores que empiezan, a conseguir escribir programas de mayor calidad, con lo que todos saldremos ganando. Por otro lado, existe la posibilidad de que los programas sean publicados en MICROHOBBY CASSETTE donde, como sabrán nuestros lectores, la recompensa puede superar ampliamente las 15.000 pesetas.

En cualquier caso, nuestra intención es fomentar la producción de software español de calidad, así como ofrecer un estímulo para aquellos programadores que estén empezando. Por tanto, no se desanime, mándenos sus programas pero, eso sí, depúrelos al máximo, sea exigente consigo mismo y sus lectores se lo agradecerán.

Reset

¿Podría dañar en algo a mi Spectrum un interruptor de corriente entre el ordenador y la fuente de alimentación para utilizarlo como RESET?

Alejandro ALVAREZ - Salamanca

□ La instalación de un interruptor a la salida de la fuente de alimentación no perjudica, en absoluto, al ordenador; no obstante, no es un RESET demasiado perfecto; si quiere utilizarlo para esto, deje pasar uno o dos segundos desde que lo desconecte hasta que lo vuelva a conectar.

Ensambladores

Con el fin de aumentar la velocidad de mis programas, decidí comprarme un ensamblador. Lo que pasa es que no sé nada sobre ensambladores. Mis dudas, pues, son las siguientes:

¿Hace falta saber código máquina para utilizar el ensamblador?

¿Ahorra memoria?

¿Tiene las mismas ventajas un programa ensamblado que uno escrito diractamente en código máquina?

¿Es muy difícil ensamblar un programa?

José M. PEÑALVER - Murcia

□ Para utilizar un ensamblador no es necesario saber código máquina, sino, Assembler. Es frecuente confundir los conceptos de Assembler y código máquina; este último es el resultado de ensamblar un programa escrito en Assembler; pero nadie programa directamente en código máquina, sería muy trabajoso; los programas se escriben en Assembler y luego se ensamblan, bien a mano, bien mediante un ensamblador.

El lenguaje Assembler ahorra memoria y es considerablemente más rápido, pero también es considerablemente más difícil que el Basic; no obstante, para eso está nuestro CURSO DE CODIGO MAQUINA que es, en realidad, un curso de lenguaje Assembler; en él podrá resolver todas sus dudas.

DE OCASION

● VENDO juegos Atari 2600 con 2 joystick y 4 cartuchos por 14.000 ptas. También vendo órgano Casio VI Tone, por 5.000 ptas. Auriculares con radio 1.000. Contactar con Barcelona llamando al Tel. (93)2420524.

● DESEARIA escribirme con lectores para poder intercambiar trucos, información, preferiblemente del Spectrum 16K, sobre Código Máquina y Basic. Dirigirse a José Viadé. Gerona, 20. Cardedeu (Barcelona).

● VENDO Spectrum 48K, en buen estado, manuales, cables, fuente de alimentación, etc. Incluyo en el mismo precio un interface para joystick tipo Kempston todo ello por 28.000 ptas. Interesados llamar al Tel. 7184278 de Madrid (a partir de las 7 de la tarde). Preguntar por Roberto o Moisés.

● VENDO Interface joystick programable con instrucciones. Precio: 2.000 ptas. Dirigirse al Tel. (91)4468278. Preguntar por Manuel (tarde).

● USUARIOS que deseen formar un club, pueden dirigirse a Daniel Pedrera. Montal Mediterráneo, Barbate de Franco o bien llamar el Tel. 430756. Sólo admito a las personas de Cádiz.

● VENDO ZX Spectrum 16K, con cinta Horizontes, y libros de instrucciones además fuente de alimentación y cables, por sólo 20.000 ptas. Interesados llamar al Tel. (91)8900668. Preguntar por Ismael Fernández.

● HEMOS formado un club del Spectrum, para el cambio o compra de revistas, ideas, información, etc. Al asociarte un regalo sorpresa. Para más información dirigirse a Carlos Galdón Torezano. S. Anastasio, 4, 8.º A. Madrid.

● VENDO video-juego Philips GP-7400 en perfecto estado, con instrucciones por sólo 19.000 ptas. Dirigirse a Fernando Valor Climent. Virgen del Socorro, 71. Alicante.

● VENDO ZX Spectrum Plus, comprado en junio-85, junto con cable, fuente de alimentación, interface tipo Kempston con 2 joystick incorporados, cassette especial para ordenadores, 2 libros sobre programación en Basic, revistas. Todo por 42.5000 ptas. Tel. (96)3785200. Valencia.

● VENDO Spectrum 48K, en perfecto estado con todos sus cables y manuales de iniciación. Con interface joystick Ram Turbo válido para todos los sistemas y entrada para cartuchos y joystick Quick Shot I por sólo 45.000 ptas. También cambio Commodore Vic-20 en perfecto estado con los cables, manuales en inglés y español, con cassette especial Data y 3 libros de iniciación y profundización al Basic por Commodore 64. Llamar al Tel. (96)2276879 o bien escribir a Jorge Morales Gutiérrez. Cosmógrafo Ramírez, 15. Xátiva (Valencia).

● VENDO consola de videojuego Philips G-7000 en perfecto estado por 20.000 ptas. (negociables). Incluye cartuchos de 6.500 ptas. Interesados llamar al Tel. 4468788 de Madrid. Preguntar por Alberto (noches).

● COMPRO Instrucciones del procesador de textos para Spectrum 48K con garantía Investrónica, instrucciones en castellano, cinta de demostración. Libro de J. Sales Roig sobre Código Máquina, y 3 revistas «ZX» por 24.000 ptas. Frances Martínez Valdés. Vilamari, 33, Barcelona. Tel. (93)2241113.

● COMPRO libro de instrucciones en español del Spectrum 48K y libros de informática o bien lo cambio por joystick Quick Shot II con interface. Para más información: Fernando Cirero. Dr. Llorente, 107. D. Benito (Badajoz).

● DESEARIA ponerme en contacto con usuarios del Spectrum de toda España (preferiblemente de Orense), con el fin de crear un club de ordenadores, para intercambiar, trucos, ideas, información, etc. Escribir a Javier. Carrero Blanco, 27, 6.º C. El Puente (Orense) o bien llamar al Tel. (988)216896.

● CAMBIO Spectrum Plus 64K, más interface programble, cables, dos manuales y 14 revistas por un Commodore CBM-64, con sus accesorios completos. Pagaremos diferencia. Interesados llamar al Tel. (91)7060299 preguntar por Pepe.

● VENDO Spectrum 48K, en perfecto estado, con 50 revistas, 2 libros, cassette marca «Vanica» por sólo 38.000 ptas. Interesados contactar al Tel. 693)3763140.

● VENDO Interface programable II y joystick Sony muy manejable. Interesados llamar al Tel. 4612382. Preguntar por Daniel.

● ESTOY interesado en la adquisición de un programa copiador Turbo, compraría copia. Interesados escribir una carta a Juan Muñoz. Gacela, 5, 3.º 1.ª Barcelona.

● URGE vender Spectrum 48K, completamente nuevo, por sólo 25.000 pta. Manuales, cinta de demostración, con cables, un cassette Sony TC-60, 3 libros, 25 revistas. Llamar al Tel. (922)272363, preguntar por Nico.

● VENDO ZX Spectrum 16K, ampliación 48K, interface, joystick Kempton, cassette Sanyo, copiones, 4 libros Spectrum y varias revistas. Precio a covenir. Interesados llamar al Tel. (983)237183. Preguntar por Eduardo o Jesús.

● VENDO video-juego Philips, poco utilizado. Precio: 15.000 ptas. Llamar al Tel. 2187067 de Madrid.

● COMPRO manual del «Hisoft Pascal 4S» para el Spectrum o pagaría fotocopias de dicho manual. Interesados dirigirse a José Redondo Dorado. Avda. Los Pallarescos, 81, 3.º 2.ª Urb. S. Salvador. Tarragona. Tel. (977)523442.

● SI te gusta la movida de la informática del Spectrum y vives en Ubeda, comarca o provincia de Jaén en general. Contacta con la Movida Informática Ubetense. Preguntar por Club Spectrum en el Tel. 750526 o bien escribir a Rafael Merelo Guervós. Corredera, 5, 2.º, izq. Ubeda. También queremos contactar con otros clubs de España.

● VENDO ZX Spectrum Plus completamente nuevo, con manual en castellano, fuente de alimentación, cables, garantía, cinta de demostración. Interesados llamar el Tel. (96)2433230 o bien escribir a: Santa Rita, 19, Carcaixent (Valencia).

● CAMBIO ordenador ZX-81 nuevo por el ordenador PB-100 que esté en perfecto estado. Interesados escribir a Vicente Padilla Romero. S. Sebastián, 27. Torredelcampo (Jaén).

● VENDO Spectrum 48K, en perfecto estado junto con TV de 12" n b/n, cintas vírgenes de 10 m., revistas para ZX Spectrum y una grabadora Philips e tan sólo 40.000 ptas. Llamar al Tel. 259582 y preguntar por Daniel. Las Palmas de Gran Canaria.

● VENDO Spectrum 48K, por 30.000 ptas., con conexiones y cassette. Lorenzo Moreno. Verdi, 10, 3.º 3.ª Barcelona.

● VENDO Spectrum 48K, más interface 1, microdrive, impresora Seikosha GP-50 S (directamente conectable al Spectrum). Lo vendo junto o separado. Interesados contactar con Gregorio. Tel. 3536853 de Barcelona (noches).

● VENDO video-juegos Philips G-7000 con 3 cintas de juegos por 13.000 ptas. A ser posible con gente de Ciudad Real. Ponerse en contacto con Roger Marrero. Juan II, 3. Ciudad Real. Tel. (926)221454.

● VENDO Spectrum 48K, completo y en perfecto estado, por 20.000 ptas. Llamar en horas de comida al Tel. 3706441 o escribir a José Luis Ruiz. Alejandro Volta, 2, 15. 46014 Valencia.

● VENDO video-juegos «Atari», con todos sus accesorios y 5 cartuchos de regalo. Interesados llamar al Tel. (91)7860379. Preguntar por Antonio.

● ME GUSTARIA contactar con usuarios del ZX Spectrum 48K y Plus. Para el intercambio de Pokes. Interesados podéis escribir a Sergi Ballester. Fray Luis de León, 151, 1.º. Sabadell (Barcelona).

● VENDO Spectrum 48K, por necesitar un ordenador mayor. Interesados llamar al Tel. (96)3518276 de Valencia y preguntar por Antonio (19 a 22 h).

● VENDO video-juegos Atari GP-2600 con mandos, transformador y tres juegos. Todo por 20.000 ptas. (negociables). Tel. 2565987. Llamar lunes, miércoles, viernes de 6 a 8 de la tarde. Madrid.

● VENDO el siguiente lote: ZX Spectrum 48K, ZX Interface 1, Microdrive. Precio especial: 50.000 ptas. Interesados escribir a Juan Gómez Martín. Cuenca, 27. 46007 Valencia.

● VENDO video-juegos CBS Colecovisión en perfecto estado con 3 cartuchos y selector de TV, o juegos por el precio de 20.000 ptas. Contactar con Alberto Roch. Tel. 2304517. Barcelona.

● VENDO teclado profesinal para Spectrum, en perfecto estado. Precio 6.000 ptas. Tel. (93)2581493 (9 a 1 h).

● VENDO ZX Spectrum 48K, en estado impecable, con fuente de alimentación, cables, cinta de demostración, en castellano, manuales de instrucciones en castellano, joystick tipo Kempston y su interface, más 4 revistas de ordenadores. Precio muy interesante. Dirigirse a Alberto Chicote. Duero, 8. Aranda de Duero. Burgos. Tel. (947)502599.

● VENDO Spectrum, 48K, nuevo, manual español, y todo lo necesario. También vendo teclado profesional Stonechip con amplificador. Junto o separado. Muy barato. Tel. (947)603179. Preguntar por Antonio. Rda. Turia, 14, 4.º. 44002 Teruel.

● INTERCAMBIO toda clase de información, ideas, trucos, etc., para Spectrum. También vendo Spectrum Plus, a estrenar, por 30.000 ptas. Interesados escribir a Tania Rodríguez. Estación Vedra. Sta. Cruz de Ribadulla. La Coruña.

Ahora exclusiva para España **Bristish Soft.**
Instrucciones en castellano. Importados de Inglaterra. Todos los precios incluido IVA. Tiendas y distribuidores. Tel. (965) 26 35 93. Pedidos contra-reembolso. Tel. (965) 26 35 93. Disponible directamente por British Soft o en los mejores establecimientos de software.
British Soft. Rocafel, 19. Albufereta *(Alicante)*

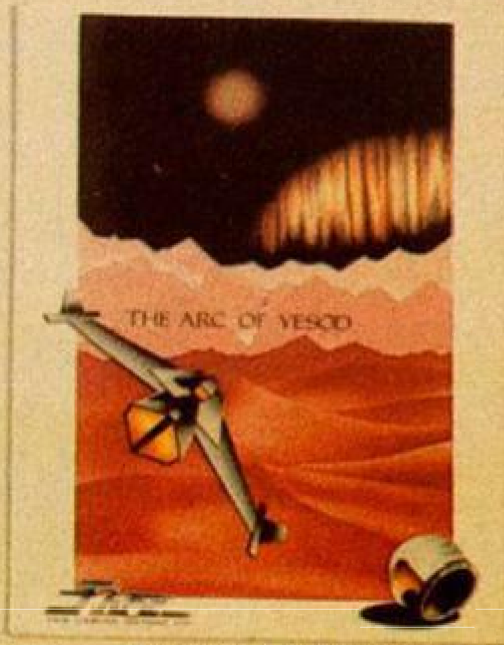
Distribuidor en Madrid:
Computique
Embajadores, 90
28012 Madrid
Tel. (91) 227 09 80



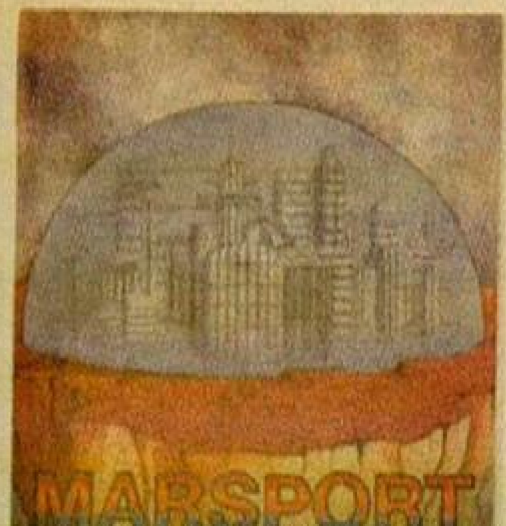
Icues: de THOR Computer Software. Nuevo. El juego más reciente de Odin. Terminado de fabricar en enero. Para Spectrum 48, 128 K. Commodore 64 ó 128 K. P.V.P. **2.300**



Robin The Wood: de Odin Computer Graphics. Crash Mash nuevo para Amstrad.
¿Puedes tú, igual que Robin ganar el trofeo más preciado, como es la flecha de plata? Para Spectrum 48, 128 K. Commodore 64 ó 128 K. Amstrad CPC. P.V.P. **2.300**



The Arc Of Yesod: Nuevo Crash Mash. De Thor Computer Software. este juego es la continuación de Nades Of Yesod. Descubrirás un mundo galáctico verdaderamente sorprendente. Spectrum 48 ó 128 K. Commodore 64 ó 128 K. Próximamente Amstrad CPC. P.V.P. **2.300**



Marsport: de Gargoyle Game. Crash Mash. Este juego es la primera parte de la trilogía. «El asedio de la tierra».
Las críticas de Microhobby «Marsport es un excelente juego sin igual». Amstrad CPC, Spectrum 48 ó 128 K. P.V.P. **2.300**



Four Pack: Amstrad CPC. De Software Proyects cuatro fantásticos juegos: Manic Miner, Binky, Jep Set Willy. Kars. Treasure Hunt. En disquette y ahora también en cassette, constituyendo una novedad sorprendente.
Disquette P.V.P. **4.000**
Cassette P.V.P. **2.300**



Sweevos World: de Gargoyle Game. Nuevo Hit. Es la más divertida aventura de dibujos animados en la que tú puedes participar. Para Amstrad CPC, Spectrum 48 ó 128 K. P.V.P. **1.900**



GP-50 •	La pequeña 40 cps. Papel normal con interface paralelo, serial y Spectrum	17.990 ptas.
GP-700 *	La de color 50 cps. 7 colores. 80 columnas. Tracción y fricción. Papel de 10 pulgadas	64.990 ptas.
SP-1.000 *	La programable 100 cps.24 cps en alta calidad 96 cart. programables en RAM. Introductor hoja a hoja.♦	64.990 ptas.
SP-1.000AS	La programable 100 cps.24 cps en alta calidad con interface RS-232. Introductor hoja a hoja.♦	59.900 ptas.
MP-1.300AI	La polivalente 300 cps, 60 cps en alta calidad, interface paralelo y RS-232. Introductor hoja a hoja.♦&	119.900 ptas.
BP-5.200 *	La de oficina 200 cps, 106 en alta calidad.Buffer 4K.Carro de 15".Tracción y fricción.♦	199.900 ptas.
BP-5.420 *	La más rápida 420 cps. 106 cps en alta calidad. Buffer de 18K. Paralelo y RS-232.♦	299.900 ptas.

Interfaces: Serie RS-232C, Spectrum, IBM, COMMODORE, MSX, QL, Apple Macintosh, HP-IB

♦ Disponen de introductor automático de documentos opcional.

& Dispone de Kit opcional de color

Nota: I.V.A. 12%, no incluido en los precios arriba indicados

* con interface paralelo

• con interface Spectrum

DiRAC

Avda. Blasco Ibáñez, 116
Tel. (96) 372.88.89
Telex 62220 - 46022 VALENCIA

Muntaner, 60-2.º-4.ª
Tel. (93) 323.32.19
08011 BARCELONA

Agustín de Foxá, 25-3.º-A
Tels. (91) 733.57.00-733.56.50
28036 MADRID